

TUGAS AKHIR - KI141502

SEGMENTASI TRABECULAR BONE PADA DENTAL PANORAMIC RADIOGRAPH MENGGUNAKAN K-MEANS CLUSTERING DAN ROOT-GUIDED DECISION TREE (RGDT)

ILHAM GURAT ADILLION
5113100077

Dosen Pembimbing I
Dr. Agus Zainal Arifin, S.Kom., M.Kom.

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

SEGMENTASI TRABECULAR BONE PADA DENTAL PANORAMIC RADIOGRAPH MENGGUNAKAN K-MEANS CLUSTERING DAN ROOT-GUIDED DECISION TREE (RGDT)

ILHAM GURAT ADILLION
5113100077

Dosen Pembimbing I
Dr. Agus Zainal Arifin, S.Kom., M.Kom.

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

TRABECULAR BONE SEGMENTATION ON DENTAL PANORAMIC RADIOGRAPH USING K-MEANS CLUSTERING AND ROOT-GUIDED DECISION TREE (RGDT)

**ILHAM GURAT ADILLION
5113100077**

**Supervisor I
Dr. Agus Zainal Arifin, S.Kom., M.Kom.**

**Supervisor II
Dini Adni Navastara, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SEGMENTASI TRABECULAR BONE PADA DENTAL PANORAMIC RADIOGRAPH MENGGUNAKAN K- MEANS CLUSTERING DAN ROOT-GUIDED DECISION TREE (RGDT)

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

ILHAM GURAT ADILLION

NRP: 5113 100 077

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Agus Zainal Arifin, S.Kom., M.Kom.
NIP. 197208091995121001

Dini Adni Navastara, S.Kom., M.Sc.
NIP. 198510172015042001



**SURABAYA
JANUARI, 2017**

[Halaman ini sengaja dikosongkan]

SEGMENTASI TRABECULAR BONE PADA DENTAL PANORAMIC RADIOGRAPH MENGGUNAKAN K-MEANS CLUSTERING DAN ROOT-GUIDED DECISION TREE (RGDT)

Nama Mahasiswa : Ilham Gurat Adillion
NRP : 5113 100 077
Jurusan : Teknik Informatika, FTIf ITS
Dosen Pembimbing 1 : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dosen Pembimbing 2 : Dini Adni Navastara, S.Kom., M.Sc.

Abstrak

Dental Panoramic Radiograph (DPR) adalah citra x-ray dua dimensi (2-D) dari gigi yang merekam keseluruhan mulut. DPR menyimpan berbagai informasi seperti struktur trabecular bone yang dapat digunakan untuk identifikasi beberapa penyakit. Namun, pada DPR, area trabecular bone cukup sulit untuk ditentukan secara akurat karena rendahnya kontras citra, pencahayaan yang tidak merata, serta banyaknya noise.

Tugas akhir ini mengusulkan sebuah metode segmentasi area trabecular bone menggunakan K-means clustering dan Root-Guided Decision Tree (RGDT). Pada Region of Interest (ROI) dari DPR dilakukan ekstraksi fitur dengan filter Gabor. Orientasi filter Gabor disesuaikan dengan orientasi dominan pada objek yang berada di dalam ROI untuk mendapatkan data fitur yang representatif terhadap objek di dalam citra. Data fitur citra kemudian dikelompokkan menggunakan K-means clustering menjadi kelas area trabecular bone dan non-trabecular bone. Adapun piksel yang keanggotaannya dalam suatu cluster dianggap ambigu karena nilai data fiturnya berbeda jauh dengan nilai data fitur titik pusat clusternya, diklasifikasi lebih lanjut menggunakan RGDT untuk menghindari kesalahan klasifikasi.

Uji coba yang dilakukan terhadap 30 citra menunjukkan bahwa metode ini dapat memberikan hasil segmentasi area

trabecular bone yang akurat dengan rata-rata akurasi, specificity dan sensitivity adalah 92,52%, 91,67%, dan 90,90%.

Kata kunci: Decision tree, dental panoramic radiograph, k-means clustering, trabecular bone.

TRABECULAR BONE SEGMENTATION ON DENTAL PANORAMIC RADIOGRAPH USING K-MEANS CLUSTERING AND ROOT-GUIDED DECISION TREE (RGDT)

Student Name : Ilham Gurat Adillion
Registration Number : 5113 100 077
Department : Informatics Engineering, FTIf ITS
First Supervisor : Dr. Agus Zainal Arifin, S.Kom.,
M.Kom.
Second Supervisor : Dini Adni Navastara, S.Kom., M.Sc.

Abstract

Dental Panoramic Radiograph (DPR) is two-dimensional (2-D) x-ray image of teeth that captures entire structure of the mouth. DPR contains many information such as trabecular bone structure that can be used to identify many diseases. But, it is hard to determine the area of trabecular bone in DPR because of low contrast, uneven lighting and high amount of noise in the image.

In this research we propose a segmentation method using K-means clustering and Root-Guided Decision Tree (RGDT). Features of Region of Interest (ROI) from DPR will be extracted using Gabor filter. Gabor filter's orientation will be adjusted with dominant orientation of objects inside ROI. The extracted features will be clustered by K-means clustering into two classes: area of trabecular bone and area of non-trabecular bone. Some pixels that have ambiguous membership in its cluster because its feature data differ too much with the cluster's centroid feature data, will be classified further using RGDT to prevent false classification.

Testing concluded to 30 image shows that this method give accurate trabecular bone area segmentation result with average accuracy, specificity and sensitivity is 92,52%, 91,67%, and 90,90%.

Keywords: Decision tree, dental panoramic radiograph, k-means clustering, trabecular bone.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**Segmentasi Trabecular Bone pada Dental Panoramic Radiograph menggunakan K-Means Clustering dan Root-Guided Decision Tree (RGDT)**”.

Buku tugas akhir ini disusun dengan harapan dapat memberikan manfaat dalam penelitian *dental panoramic radiograph* lebih lanjut. Selain itu, penulis berharap dapat memberikan kontribusi positif bagi kampus Teknik Informatika ITS.

Dalam perancangan, pengerjaan, dan penyusunan tugas akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Bapak Dr. Agus Zainal Arifin, S.Kom., M.Kom. dan Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
2. Orang tua penulis Bapak Djoko Sutikno dan Almh. Ibu Sutiwi yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
3. Seluruh saudara kandung: tiga kakak (mbak Sisca, mbak Shinta, mbak Shilvy) dan keluarga (mas Armand, mas Heri; Maira, Maura, Nisa, Saiba, Ahmad) serta seluruh keluarga besar yang telah memberikan dukungan yang besar baik secara langsung maupun secara implisit.
4. Teman-teman di Lab KCV: para admin yang telah banyak membantu memfasilitasi dalam pengerjaan serta teman-

teman yang berhabitat hidup di KCV, yang sudah menemani penulis.

5. Teman satu bimbingan Bu Dini (Saddam dan Putri) dan teman satu bimbingan Pak Agus (Rizok) yang sering bimbingan bersama, serta seluruh anggota zemi Pak Agus.
6. Teman-teman seperti Rahmat, Rijal, Bandem, dan mahasiswa angkatan 2013 lain yang sama-sama mengarungi empat tahun masa perkuliahan bersama penulis, yang saat berinteraksi dan diam-diam mengamati tingkah laku mereka, telah memberi penulis banyak pelajaran hidup secara tersirat maupun tersurat dengan gratis.
7. Pihak-pihak lain yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Januari 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Tugas Akhir	3
1.5 Manfaat Tugas Akhir	4
1.6 Metodologi	4
1.7 Sistematika Laporan.....	5
BAB II DASAR TEORI.....	7
2.1 <i>Dental Panoramic Radiograph</i>	7
2.1.1 Pengertian <i>Dental Radiograph</i>	8
2.1.2 Teori <i>Panoramic Radiographs</i>	8
2.2 Struktur Penyusun <i>Dental Panoramic Radiograph</i>	10
2.2.1 Tulang.....	10
2.2.2 Cortical Bone.....	11
2.2.3 Trabecular Bone	11
2.3 <i>Region of Interest (ROI)</i>	13
2.4 Normalisasi	13
2.4.1 Normalisasi Skala	13
2.4.2 Normalisasi Skala Rentang Baru	13
2.5 Morfologi	14
2.5.1 <i>Structuring Element</i>	14
2.5.2 Operator <i>Opening</i>	15

2.6	Filter Citra	16
2.6.1	<i>Mean Filter</i>	16
2.6.2	<i>Median Filter</i>	17
2.6.3	<i>Motion Blur Filter</i>	17
2.6.4	<i>Gabor Filter</i>	19
2.7	K-means Clustering	21
2.8	Decision Tree	22
2.8.1	Algoritma C4.5	23
2.8.2	Root-Guided Decision Tree	24
2.9	Confusion Matrix	24
BAB III ANALISIS DAN PERANCANGAN		27
3.1	Tahap Analisis	27
3.1.1	Deskripsi Umum	27
3.1.2	Spesifikasi Kebutuhan Sistem	27
3.1.3	Analisis Permasalahan	28
3.1.4	Analisis Permasalahan <i>Clustering</i>	31
3.1.5	Analisis Permasalahan Klasifikasi	31
3.1.6	Analisis Permasalahan <i>Postprocessing</i>	32
3.2	Tahap Perancangan	32
3.2.1	Perancangan Sistem	33
3.2.2	Perancangan Data	36
3.2.3	Perancangan Proses	36
BAB IV IMPLEMENTASI		49
4.1	Lingkungan Implementasi	49
4.1.1	Perangkat Keras	49
4.1.2	Perangkat Lunak	49
4.2	Implementasi Tahap <i>Preprocessing</i>	50
4.2.1	Implementasi Pembagian Citra	50
4.2.2	Implementasi Normalisasi Citra	50
4.2.3	Implementasi <i>Filtering</i>	51
4.3	Implementasi Ekstraksi Fitur	51
4.3.1	Implementasi Gabor Filter	52
4.3.2	Implementasi Pengambilan Fitur Spasial	53

4.3.3 Implementasi Pengambilan Koordinat Sumbu Y	54
4.4 Implementasi Proses <i>Clustering</i>	54
4.4.1 Implementasi Proses <i>K-means Clustering</i>	54
4.4.2 Implementasi Penghitungan Jarak Piksel ke <i>Centroid</i> dan <i>Thresholding</i>	56
4.5 Implementasi Proses Klasifikasi	58
4.5.1 Implementasi <i>Training Decision Tree</i>	58
4.5.2 Implementasi Klasifikasi Ulang Piksel.....	61
4.5.3 Implementasi <i>Postprocessing</i>	62
BAB V Uji Coba dan Evaluasi	63
5.1 Lingkungan Uji Coba.....	63
5.2 Data Uji Coba.....	63
5.3 Skenario Uji Coba.....	65
5.4 Uji Coba Penentuan Parameter pada <i>Gabor Filter</i>	66
5.4.1 Uji Coba Penentuan Parameter Sudut pada <i>Gabor Filter</i> ..	66
5.4.2 Uji Coba Penentuan Parameter <i>Wavelength</i> pada <i>Gabor Filter</i>	68
5.4.3 Uji Coba Penentuan Parameter <i>Bandwith</i> pada <i>Gabor Filter</i>	69
5.4.4 Uji Coba Penentuan Parameter <i>Gamma</i> pada <i>Gabor Filter</i> ..	71
5.5 Uji Coba Fitur Spasial.....	73
5.6 Uji Coba Penentuan Nilai <i>Threshold</i> pada Hasil <i>K-means Clustering</i>	75
5.7 Uji Coba Level <i>Pruning</i> Pada RGDT	77
5.8 Uji Coba Metode Pembandingan	78
5.9 Evaluasi.....	81
5.9.1 Evaluasi Uji Coba Parameter Filter Gabor	81
5.9.2 Evaluasi Uji Coba Fitur Spasial.....	83
5.9.3 Evaluasi Uji Coba Nilai <i>Threshold</i>	84
5.9.4 Evaluasi Uji Coba Level <i>Pruning</i> pada <i>Decision Tree</i>	85
5.9.5 Evaluasi Uji Coba Metode Pembandingan.....	85
5.9.6 Evaluasi Hasil Segmentasi Metode	86
BAB VI Kesimpulan dan Saran.....	91

6.1	Kesimpulan	91
6.2	Saran.....	92
LAMPIRAN.....		93
BIODATA PENULIS.....		115

DAFTAR GAMBAR

Gambar 2.1 <i>Dental Panoramic Radiograph</i>	7
Gambar 2.2 Posisi Pasien dalam Pengambilan <i>Panoramic Radiograph</i>	9
Gambar 2.3 Hasil Foto <i>Panoramic Radiograph</i> Tampak Samping	9
Gambar 2.4 Citra <i>Dental Panoramic Radiograph</i> dengan 2 ROI	10
Gambar 2.5 <i>Trabecular Bone</i> dan <i>Cortical Bone</i> pada Rahang..	12
Gambar 2.6 Struktur <i>Cortical Bone</i>	12
Gambar 2.7 Struktur <i>Trabecular Bone</i>	12
Gambar 2.8 <i>Structuring element</i>	15
Gambar 2.9 Operasi Dilasi	15
Gambar 2.10 Operasi Erosi	16
Gambar 2.11 Operasi <i>Opening</i>	16
Gambar 2.12 <i>Kernel mean filter</i>	18
Gambar 2.13 <i>Median filtering</i>	18
Gambar 2.14 <i>Kernel motion blur</i>	18
Gambar 2.15 Hasil <i>Motion Blur</i>	18
Gambar 2.16 <i>Gabor Filter</i>	19
Gambar 2.17 Hasil filter gabor	20
Gambar 2.18 Ilustrasi <i>K-means Clustering</i>	21
Gambar 2.19 Struktur <i>decision tree</i>	22
Gambar 2.20 <i>Confusion Matrix</i> berdasar Area.....	25
Gambar 3.1 ROI bawah.....	29
Gambar 3.2 Detail Struktur <i>Trabecular Bone</i> yang terdapat diantara dua akar gigi	29
Gambar 3.3 Pengaruh orientasi pada filter Gabor	30
Gambar 3.4 Contoh Area-Area Terang pada ROI.....	33
Gambar 3.5 Ilustrasi Pengambilan Foto Rahang	34
Gambar 3.6 Ilustrasi Proses <i>Scanning</i> Foto.....	34
Gambar 3.7 Diagram Alir dari Sistem Aplikasi Segmentasi area <i>Trabecular Bone</i>	35
Gambar 3.8 Citra ROI	38
Gambar 3.9 Citra Hasil Segmentasi	38
Gambar 3.10 Diagram Alir Tahap <i>Preprocessing</i>	39

Gambar 3.11 Pemotongan ROI	39
Gambar 3.12 Hasil <i>preprocessing</i>	40
Gambar 3.13 Diagram Alir Proses Ekstraksi Fitur	41
Gambar 3.14 Hasil <i>Gabor filter</i>	42
Gambar 3.15 Hasil <i>Clustering</i> dan <i>Thresholding</i>	43
Gambar 3.16 Diagram Alir Proses <i>Clustering</i>	44
Gambar 3.17 Hasil Klasifikasi.....	45
Gambar 3.18 Diagram Alir Proses Klasifikasi	45
Gambar 3.19 Hasil <i>postprocessing</i>	46
Gambar 3.20 Diagram Alir Proses Postprocessing.....	47
Gambar 5.1 Citra <i>Dental Panoramic Radiograph</i>	64
Gambar 5.2 Citra Hasil <i>Cropping</i>	64
Gambar 5.3 Citra <i>Ground Truth</i>	65
Gambar 5.4 Hasil Uji Coba Parameter <i>Wavelength</i> pada ROI Atas	69
Gambar 5.5 Hasil Uji Coba Parameter <i>Wavelength</i> pada ROI Bawah.....	70
Gambar 5.6 Hasil Uji Coba Parameter <i>Bandwith</i> pada ROI Atas.....	71
Gambar 5.7 Hasil Uji Coba Parameter <i>Bandwith</i> pada ROI Bawah	72
Gambar 5.8 Hasil Uji Coba Parameter <i>Gamma</i> pada ROI Atas.....	73
Gambar 5.9 Hasil Uji Coba Parameter <i>Gamma</i> pada ROI Bawah	74
Gambar 5.10 Hasil Uji Coba Fitur Spasial pada ROI Atas	75
Gambar 5.11 Hasil Uji Coba Fitur Spasial pada ROI Atas	76
Gambar 5.12 ROI Atas terburuk.....	87
Gambar 5.13 ROI Bawah terburuk.....	88

DAFTAR TABEL

Tabel 5.1 Hasil Uji Coba Parameter Sudut	67
Tabel 5.2 Hasil Uji Coba Parameter <i>Wavelength</i> pada ROI Atas	68
Tabel 5.3 Hasil Uji Coba Parameter <i>Wavelength</i> pada ROI Bawah	69
Tabel 5.4 Hasil Uji Coba Parameter <i>Bandwith</i> pada ROI Atas...	70
Tabel 5.5 Hasil Uji Coba Parameter <i>Bandwith</i> pada ROI Bawah	71
Tabel 5.6 Hasil Uji Coba Parameter <i>Gamma</i> pada ROI Atas	72
Tabel 5.7 Hasil Uji Coba Parameter <i>Gamma</i> pada ROI Bawah..	73
Tabel 5.8 Hasil Uji Coba Fitur Spasial pada ROI Bawah	74
Tabel 5.9 Hasil Uji Coba Fitur Spasial pada ROI Atas	75
Tabel 5.10 Hasil Uji Coba Nilai <i>Threshold</i> pada ROI Atas	77
Tabel 5.11 Hasil Uji Coba Nilai <i>Threshold</i> pada ROI Bawah ...	77
Tabel 5.12 Hasil Uji Coba Level <i>Pruning</i>	78
Tabel 5.13 Hasil Uji Coba Metode Pembanding pada ROI Atas	79
Tabel 5.14 Hasil Uji Coba Metode Pembanding pada ROI Bawah	80
Tabel 5.15 Hasil segmentasi pada ROI	86

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Pembagian Citra	50
Kode Sumber 4.2 Implementasi Normalisasi Citra	51
Kode Sumber 4.3 Implementasi <i>Filtering</i>	51
Kode Sumber 4.4 Implementasi Perulangan Orientasi <i>Gabor Filter</i>	52
Kode Sumber 4.5 Implementasi Pembentukan <i>Gabor Filter</i>	53
Kode Sumber 4.6 Implementasi Pengambilan Fitur Spasial	54
Kode Sumber 4.7 Implementasi Pengambilan Koordinat Sumbu Y	54
Kode Sumber 4.8 Implementasi <i>K-means Clustering</i> dan Pemberian Warna ROI Atas	55
Kode Sumber 4.9 Implementasi <i>K-means Clustering</i> dan Pemberian Warna ROI Bawah	56
Kode Sumber 4.10 Implementasi Inisialisasi <i>Centroid</i>	56
Kode Sumber 4.11 Implementasi Penghitungan Jarak Centroid dan <i>Thresholding</i>	57
Kode Sumber 4.12 Implementasi Penyusunan <i>Tree</i> untuk Tiap Atribut	59
Kode Sumber 4.13 Implementasi Penyusunan <i>Tree</i> secara Khusus	61
Kode Sumber 4.14 Implementasi Klasifikasi Ulang Piksel	62
Kode Sumber 4.15 Implementasi <i>Postprocessing</i>	62

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari tugas akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir.

1.1 Latar Belakang

Dental panoramic radiograph (DPR) adalah citra *x-ray* dua dimensi (2-D) dari gigi yang merekam keseluruhan mulut, termasuk gigi, rahang atas, rahang bawah dan jaringan serta struktur yang melingkupinya dalam satu citra. DPR adalah citra yang mudah dan murah diperoleh sehingga sering digunakan sebagai alat bantu diagnosa penyakit-penyakit serius seperti osteoporosis [1]. DPR memiliki berbagai informasi yang terekam di dalamnya, seperti struktur *trabecular bone* yang dapat digunakan untuk identifikasi berbagai penyakit.

Pada DPR, area *trabecular bone* tidak mudah untuk diamati karena rendahnya kontras citra, pencahayaan yang tidak merata serta banyaknya *noise*. Oleh karena itu, pada DPR perlu dilakukan pengolahan citra terlebih dahulu untuk mendapatkan area *trabecular bone* yang akurat. Pengolahan tersebut adalah proses segmentasi. Segmentasi dilakukan untuk membedakan antara area *trabecular bone* dan bagian lain dari DPR. Beberapa penelitian telah dilakukan mengenai segmentasi area *trabecular bone* oleh Arifin, dkk [2] menggunakan informasi tekstur dan Arifin, dkk [3] menggunakan *linear structure*.

Segmentasi citra yang baik memerlukan data fitur yang dapat merepresentasikan bentuk objek dalam citra baik menurut bentuk, orientasi maupun ketetanggaan piksel. Filter Gabor adalah filter yang banyak digunakan dalam mendapatkan informasi tekstur dan mampu memberikan nilai yang tinggi pada objek yang memiliki orientasi searah dengan orientasi filter [4]. Selain filter Gabor,

informasi spasial pada citra dapat memberikan data fitur yang representatif terhadap bentuk objek [5].

Segmentasi dapat dilakukan dengan cara *supervised learning* dan *unsupervised learning*. Salah satu metode *unsupervised learning* yang paling umum digunakan dalam proses *clustering* adalah *K-means clustering* [6]. Sedangkan untuk *supervised learning*, pada penelitian Fraz, dkk [7] *decision tree* digunakan untuk mensegmentasi citra dan mendapatkan hasil yang cukup akurat. GeethaRamani dan Balasubramanian [8] dalam penelitiannya menggabungkan metode *K-means clustering* sebagai *unsupervised learning* dan *Root-Guided Decision Tree* (RGDT) sebagai *supervised learning*. Penggabungan ini diterapkan sebagai cara menghindari kesalahan klasifikasi yang dapat terjadi pada *K-means clustering* karena ukuran *cluster* yang sama. Hal ini menimbulkan ambiguitas keanggotaan piksel yang nilai data fiturnya berbeda jauh dengan nilai data fitur titik pusat *clusternya*. Untuk mengatasi kekurangan tersebut diperlukan sebuah *classifier* untuk mengevaluasi keanggotaan piksel yang ambigu. Proses evaluasi ini memerlukan *classifier* dengan akurasi yang tinggi sehingga tidak memperburuk hasil segmentasi yang didapat dari *K-means clustering*. RGDT membangun kumpulan *tree* dengan masing-masing atribut sebagai *node*. Kemudian dipilih *tree* dengan nilai akurasi terbaik, sehingga hasil *decision tree* yang didapat merupakan *decision tree* dengan akurasi tertinggi [8].

Pada area *trabecular bone* dalam DPR terdapat struktur *trabecular bone* yang intensitas pikselnya mirip dengan objek non-*trabecular bone* seperti akar gigi dan *cortical bone*. Hal ini mengakibatkan rendahnya akurasi apabila dilakukan *thresholding* berdasarkan nilai intensitas piksel saja. Penggabungan metode *K-means clustering* dan RGDT [8] sangat mungkin diterapkan dalam segmentasi area *trabecular bone* karena dapat mensegmentasi citra dengan data fitur yang representatif terhadap area *trabecular bone*.

Pada tugas akhir ini akan diterapkan penggabungan metode *K-means clustering* dan RGDT untuk melakukan segmentasi area *trabecular bone* pada DPR. Hasil filter Gabor dengan orientasi

yang sesuai dengan orientasi dominan objek dalam ROI serta penghitungan informasi spasial sebagai data masukan *K-means clustering* diharapkan mampu memberikan data fitur yang representatif terhadap objek dalam citra. Penerapan RGDT untuk mengevaluasi piksel ambigu pada hasil *K-means clustering* diharapkan dapat meningkatkan performa segmentasi sehingga dihasilkan area *trabecular bone* yang akurat.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana melakukan ekstraksi fitur pada DPR agar mendapat data fitur yang representatif terhadap objek dalam citra?
2. Bagaimana mengklasifikasi piksel dari data fitur sebagai area *trabecular bone* atau area non-*trabecular bone*?
3. Bagaimana menghindari kesalahan klasifikasi piksel dari data fitur?
4. Bagaimana menghitung performa hasil segmentasi area *trabecular bone*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Dilakukan pemilihan ROI secara manual sebagai *input* program dengan ukuran 256 x 128 piksel.
2. ROI berupa citra *grayscale* yang mencakup area *trabecular bone* pada DPR sebanyak 30 buah.
3. Metode diimplementasikan menggunakan MATLAB.

1.4 Tujuan Tugas Akhir

Tujuan tugas akhir ini adalah melakukan segmentasi *trabecular bone* pada DPR menggunakan metode *K-means clustering* dan *Root-Guided Decision Tree* (RGDT).

1.5 Manfaat Tugas Akhir

Manfaat dari tugas akhir ini adalah menghasilkan sistem untuk mensegmentasi area *trabecular bone* pada DPR secara akurat. Hasil segmentasi area *trabecular bone* dapat dijadikan dasar dalam perhitungan informasi struktur *trabecular bone*.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Pada studi literatur, dilakukan pengumpulan data dan studi terhadap sejumlah referensi yang diperlukan dalam pengerjaan tugas akhir. Referensi tersebut didapatkan dari beberapa artikel yang dipublikasikan oleh jurnal. Selain dari artikel, studi literatur juga dilakukan melalui pencarian referensi dari internet yang membahas mengenai informasi yang dibutuhkan, seperti DPR, *Gabor filter*, *K-means clustering*, *decision tree*, dan beberapa operasi morfologi.

2. Analisis dan Desain Perangkat Lunak

Pada tahap ini disusun rancang bangun dari perangkat lunak yang dibangun. Pengguna dapat memilih ROI dari DPR berukuran 256 x 128 piksel yang telah disediakan sebagai data masukan. Kemudian, sistem akan memproses ROI dengan melakukan *preprocessing*, ekstraksi fitur, *clustering*, klasifikasi, dan *postprocessing*. Setelah proses selesai, sistem akan menampilkan citra area *trabecular bone* yang telah tersegmentasi.

3. Implementasi Perangkat Lunak

Sistem segmentasi area *trabecular bone* pada DPR akan dibuat dengan bahasa pemrograman MATLAB menggunakan kakas bantu IDE MATLAB 8.3 (R2014a) pada platform *desktop*. *Toolbox* yang digunakan untuk mendukung pengerjaan adalah *image processing toolbox* dan *statistics and machine learning toolbox*. Kakas bantu pendukung lain diantaranya *GIMP* untuk

membuat *ground truth*, *Microsoft Paint* untuk dokumentasi, dan *Microsoft Excel* sebagai pengolah angka.

4. Uji Coba dan Evaluasi

Dalam tahap ini, dilakukan pengujian parameter-parameter yang dibutuhkan pada proses ekstraksi fitur, *clustering* dan klasifikasi. Citra keluaran yang telah tersegmentasi dibandingkan dengan citra *ground truth* untuk dihitung performanya dengan menghitung nilai akurasi, *specificity* dan *sensitivity*.

1.7 Sistematika Laporan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini berisi tentang analisis dan perancangan desain sistem segmentasi area *trabecular bone*.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini membahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk kinerja dari aplikasi yang dibangun.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan aplikasi ke depannya.

BAB II

DASAR TEORI

Pada bab ini diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir dengan tujuan untuk memberikan gambaran secara umum terhadap penelitian yang dikerjakan. Bab ini berisi penjelasan mengenai *dental panoramic radiograph*, *trabecular bone* dan *Region of Interest (ROI)* sebagai ruang lingkup dan studi kasus, metode *Gabor filter* untuk ekstraksi fitur, *K-means clustering* dalam proses *clustering*, metode *Root-Guided Decision Tree* untuk klasifikasi, serta morfologi untuk memperbaiki segmen citra biner.

2.1 *Dental Panoramic Radiograph*

Dental Panoramic Radiograph (DPR) merupakan foto *x-ray* yang mencakup keseluruhan struktur mulut manusia. DPR menyimpan informasi berupa susunan dan jaringan tulang, gigi serta bahan lain seperti gigi palsu atau aksesoris gigi lain.

Citra DPR adalah objek citra yang akan diolah pada tugas akhir ini. Contoh DPR ditunjukkan pada **Gambar 2.1**.



Gambar 2.1 *Dental Panoramic Radiograph*

2.1.1 Pengertian *Dental Radiograph*

Radiography adalah sebuah proses untuk merekam struktur bagian dalam tubuh manusia dalam sebuah citra statis. Pada proses *radiography*, sinar *x-ray* dipancarkan melewati tubuh. Sebagian sinar *x-ray* terserap atau terpecah karena mengenai struktur dalam tubuh. Sinar *x-ray* yang tersisa diterima oleh detektor dan disimpan sebagai sebuah citra [9].

Radiography banyak digunakan sebagai alat untuk mendiagnosa penyakit yang dapat dideteksi lewat citra statis. Beberapa contoh penggunaannya antara lain untuk pemeriksaan gigi, verifikasi lokasi pembedahan, *mammography*, evaluasi ortopedi, dan lain-lain [9].

Dental radiograph atau *dental x-ray* adalah citra yang menyimpan informasi berupa gigi, tulang dan jaringan-jaringan lunak pada mulut. *Dental radiograph* dapat mendeteksi struktur *dental* tersembunyi seperti gigi bungsu, kawah gigi, dan pengeroposan tulang sehingga sering digunakan untuk mendeteksi masalah gigi [10].

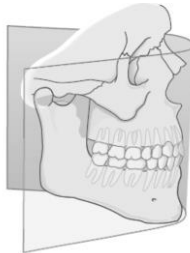
Dental radiograph dibentuk dengan mengatur radiasi sinar *x-ray*. Sinar *x-ray* menembus struktur gigi sampai mengenai sensor film. Struktur gigi yang padat akan terlihat jelas/terang karena radiasi yang mencapai struktur film lebih sedikit. Kerusakan gigi, infeksi, maupun karies yang kurang padat akan terlihat gelap karena sinar *X-ray* lebih dapat menembusnya sampai ke sensor film.

2.1.2 Teori *Panoramic Radiographs*

Panoramic Radiography atau *panoramic x-ray* adalah metode pemeriksaan gigi yang menyimpan struktur keseluruhan mulut dalam sebuah citra dua dimensi (2-D). Struktur yang tersimpan di antaranya adalah gigi, rahang atas, rahang bawah dan keseluruhan jaringan yang ada. Rahang adalah tulang yang berbentuk melengkung seperti tapal kuda, namun pada *panoramic radiograph* rahang disimpan dalam bentuk mendatar [11].

Proses pengambilan *panoramic radiograph* dilakukan dengan memutar sebuah tabung *x-ray*. Tabung tersebut berputar sampai setengah lingkaran pada kepala pasien, mulai dari salah satu ujung sisi rahang ke ujung sisi rahang yang lain. *Panoramic radiograph* tidak menggunakan film di dalam mulut pasien, melainkan, sinar *x-ray* yang melewati kepala pasien ditangkap oleh sensor film yang ikut berputar di kepala pasien di sisi yang berlawanan dengan tabung *x-ray* [11]. Posisi pengambilan *panoramic radiograph* ditunjukkan pada **Gambar 2.2**

Saat ini, *panoramic radiograph* tersimpan dalam bentuk file digital secara elektronik. Penyimpanan dalam bentuk digital ini memberi keuntungan karena mudah diakses untuk diagnosa, juga memungkinkan dokter gigi untuk memanipulasi citra *panoramic radiograph* baik dari sisi kontras, kecerahan, dan lain-lain untuk memperjelas struktur atau jaringan tertentu yang ada di dalamnya [11]. Contoh hasil foto *panoramic radiograph* ditunjukkan pada **Gambar 2.3**



Gambar 2.2 Posisi Pasien dalam Pengambilan *Panoramic Radiograph*

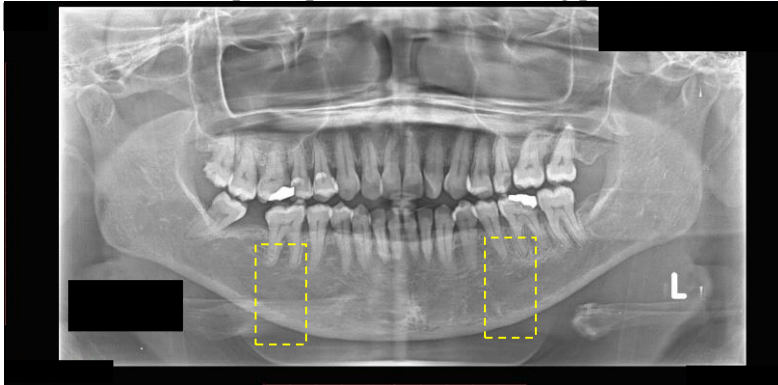


Gambar 2.3 Hasil Foto *Panoramic Radiograph* Tampak Samping

2.2 Struktur Penyusun *Dental Panoramic Radiograph*

Dalam *dental panoramic radiograph* tersimpan berbagai bentuk objek seperti yang ditunjukkan oleh **Gambar 2.1** Objek-objek yang terdapat pada *Dental Panoramic Radiograph* diantaranya adalah gigi, *trabecular bone*, dan *cortical bone*.

Region of Interest (ROI) *dental panoramic radiograph* yang diambil pada tugas akhir ini adalah bagian *trabecular bone*, ditandai oleh kotak putus-putus berwarna kuning pada **Gambar 2.4**.



Gambar 2.4 Citra *Dental Panoramic Radiograph* dengan 2 ROI

2.2.1 Tulang

Tulang adalah jaringan tubuh padat yang tersusun dari deretan sel pada material keras. Tulang memiliki dua komponen utama yaitu kolagen dan kalium fosfat. Fungsi dari tulang adalah untuk mendukung gerakan jaringan lunak seperti kontraksi otot dan mengembangkannya paru-paru, melindungi organ lunak dan jaringan seperti tengkorak, menyediakan lokasi jaringan unik seperti sumsum tulang dan penyimpanan mineral, sebagai tempat sistem endokrin meregulasi level sirkulasi kalsium dan fosfat [12].

Setiap tulang mempunyai lapisan luar (*cortex*) yang mulus, kompak, dan mempunyai ketebalan bervariasi. Bagian ini disebut *cortical bone*. Di dalamnya, terdapat sebuah jaringan saling silang berupa serabut yang disebut *trabeculae*, yang jumlahnya bervariasi

sesuai lokasi dan kondisi. Jaringan seperti serabut ini disebut *trabecular bone* [12]. Posisi *cortical bone* dan *trabecular bone* pada rahang manusia ditunjukkan pada **Gambar 2.5**.

2.2.2 Cortical Bone

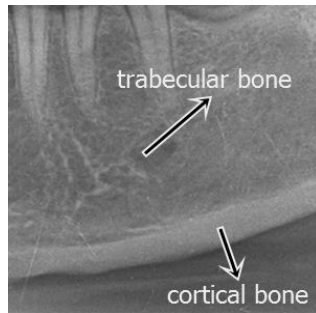
Cortical bone atau *compact bone* adalah bagian padat pada tulang. Pada *cortical bone* susunan tulang sangat padat dengan komposisi organik dan anorganik, dengan sel tulang atau *osteocytes* mengisi sisanya. *Cortical bone* menyusun sekitar 80% dari seluruh tulang manusia [13]. Struktur *trabecular bone* ditunjukkan pada **Gambar 2.6**

Cortical bone melindungi bagian lunak tulang (*trabecular bone*) dengan strukturnya yang keras dan merupakan bagian paling umum pada tulang kaki dan tangan. *Cortical bone* matang tersusun secara berlapis dan konsentris di sekitar kanal pembuluh darah, membentuk struktur *osteon*. *Cortical bone* yang belum matang tidak mempunyai *osteon* [13].

2.2.3 Trabecular Bone

Trabecular bone, *spongy bone* atau *cancellous bone* adalah bagian tulang yang ringan, berongga dalam bentuk spons atau sarang lebah sehingga dapat mengisi ruang yang sangat besar. Susunan tulang ini terdiri dari serabut-serabut tulang yang disebut *trabeculae*, di mana rongga-rongga di tengahnya biasa diisi dengan sumsum atau pembuluh darah [14]. Struktur *trabecular bone* ditunjukkan pada **Gambar 2.7**.

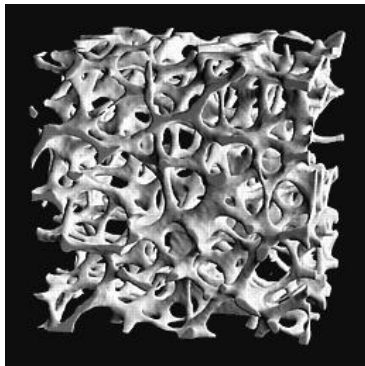
Trabecular bone menyusun sekitar 20 persen dari keseluruhan rangka tulang dalam manusia. Tulang ini ditemukan pada daerah ujung tulang (*epiphyses*), rusuk, pundak, dan tengkorak. Tulang ini berada di area yang tidak terkepos tekanan atau beban yang berat. Ukuran rongga-rongga yang ada di dalam *trabecular bone* bervariasi pada tulang-tulang yang ada bergantung kebutuhan tingkat kekuatan dan fleksibilitas tulang tersebut. Pada *trabecular bone* juga terjadi banyak aktivitas metabolisme level tinggi [14].



Gambar 2.5 *Trabecular Bone* dan *Cortical Bone* pada Rahang



Gambar 2.6 Struktur *Cortical Bone*



Gambar 2.7 Struktur *Trabecular Bone*

2.3 *Region of Interest (ROI)*

Region of Interest (ROI) adalah potongan citra pada suatu area tertentu, dimana akan dilakukan *filtering* atau operasi lain [11]. ROI umumnya diambil untuk mengamati karakteristik morfologis citra atau mengekstrak data spesifik dari citra. Informasi yang didapat dari ROI bisa berupa volume, panjang, dimensi, atau titik pusat [16]. Pengambilan ROI pada DPR ditunjukkan oleh **Gambar 2.4**.

Pemilihan ROI pada tugas akhir ini dilakukan pada area rahang bawah yang mencakup bagian *trabecular bone*. Bagian atas dari ROI yang dipilih dibatasi oleh akar gigi dan bagian bawah dibatasi oleh *cortical bone*.

2.4 Normalisasi

Normalisasi adalah teknik untuk menstandarkan atau menyamakan rentang data sehingga tidak ada satu atribut yang terlalu dominan atas atribut yang lain. Pada tugas akhir ini, normalisasi dilakukan sekali, yaitu normalisasi *region of interest*.

2.4.1 Normalisasi Skala

Normalisasi skala [17] adalah teknik penstandaran data pada suatu rentang tertentu, umumnya 0-1. Diketahui nilai maksimal dari data fitur sebagai X_{max} dan nilai minimal dari data fitur adalah X_{min} . Data yang akan dinormalisasi disimbolkan dengan X .

Nilai hasil normalisasi yang dilambangkan dengan huruf Y didapatkan dari rumus perhitungan normalisasi skala ditunjukkan pada persamaan (2.1) berikut:

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}}. \quad (2.1)$$

2.4.2 Normalisasi Skala Rentang Baru

Normalisasi skala rentang baru [17] adalah variasi dari perhitungan normalisasi skala. Pada normalisasi skala rentang baru, diberikan sebuah rentang baru sebagai acuan. Pada tugas

akhir ini, normalisasi skala rentang baru digunakan untuk standardisasi nilai intensitas piksel citra.

Pada citra, rentang baru yang digunakan adalah nilai maksimal dan minimal intensitas piksel citra yaitu 0 dan 255, yang disimbolkan dengan $X_{2\min}$ dan $X_{2\max}$. Intensitas piksel maksimum dan minimum dari citra yang akan dinormalisasi disimbolkan dengan $X_{1\min}$ dan $X_{1\max}$, Intensitas piksel citra sebelum normalisasi dilambangkan dengan X_I , sedangkan intensitas piksel citra hasil normalisasi dilambangkan dengan Y . Y didapatkan dari rumus perhitungan normalisasi skala rentang baru yang ditunjukkan pada persamaan (2.2) berikut:

$$Y = \frac{X_1 - X_{1\min}}{X_{1\max} - X_{1\min}} \times (X_{2\max} - X_{2\min}) \quad (2.2)$$

2.5 Morfologi

Operasi morfologi adalah operasi yang berkaitan dengan komponen citra yang dapat mendeskripsikan bentuk objek pada citra, seperti batas objek, rangka objek dan *convex hull*. Operasi morfologi biasa dilakukan untuk *preprocessing* citra, diantaranya adalah *filtering*, *thinning* dan *pruning* [18].

Pada operasi morfologi, sebuah himpunan bilangan digunakan untuk mendeskripsikan citra. Pada morfologi dilakukan kombinasi antara matriks citra dan *structuring element* dengan operasi matematika seperti interseksi, union, inklusi atau komplemen.

2.5.1 Structuring Element

Structuring element atau biasa disebut dengan *kernel* adalah kumpulan pola spesifik berupa koordinat angka diskrit. Representasi koordinat angka yang biasa digunakan adalah koordinat *cartesian*. *Structuring element* kemudian diputar dan dikombinasikan dengan nilai citra yang dilewatinya.

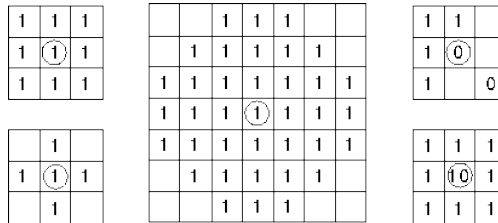
Structuring element banyak digunakan pada operasi umum pada *image processing*, diantaranya operasi erosi, dilasi, *opening*

dan *closing* [19]. Bentuk *structuring element* ditunjukkan pada **Gambar 2.8**

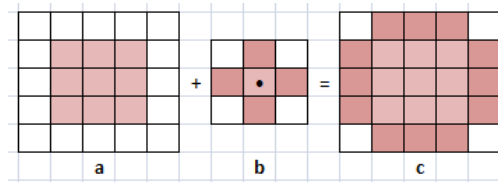
2.5.2 Operator Opening

Operator *opening* adalah gabungan operasi erosi dan dilasi. Operator dilasi adalah morfologi citra yang digunakan untuk memperbesar batas dari sebuah objek pada citra, sehingga ukuran dari objek tersebut membesar. Sedangkan operator erosi adalah morfologi citra yang digunakan untuk menghapus batas dari sebuah objek pada citra, sehingga ukuran dari objek tersebut mengecil. Operator erosi dan dilasi masing-masing ditunjukkan pada **Gambar 2.9** dan **Gambar 2.10**.

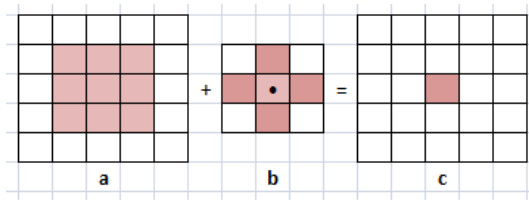
Operator *opening* dilakukan untuk menipiskan batas dari suatu objek pada citra namun tidak sebanyak operasi erosi. Operator *opening* membutuhkan sebuah *structuring element* untuk menentukan arah operasi. Operator *opening* adalah operasi erosi yang dilanjutkan dengan dilasi, sehingga mempunyai efek erosi namun tetap menjaga bentuk *foreground* yang sama dengan *structuring element* yang dipakai [20]. Contoh penerapan operasi *opening* ditunjukkan pada **Gambar 2.11**



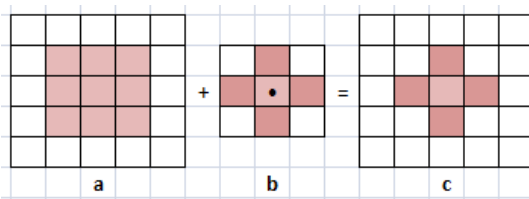
Gambar 2.8 Structuring element



Gambar 2.9 Operasi Dilasi



Gambar 2.10 Operasi Erosi



Gambar 2.11 Operasi Opening

2.6 Filter Citra

Filter citra digunakan untuk mereduksi frekuensi tinggi pada citra seperti *smoothing* atau frekuensi rendah seperti deteksi tepi. Beberapa contoh dari filter citra adalah *mean filter*, *median filter*, *gaussian filter* dan *gabor filter*.

Filter terbagi menjadi filter frekuensi dan filter spasial. Pada filter frekuensi citra dirubah kedalam domain frekuensi dan dikalikan dengan filter frekuensi sebelum dikembalikan ke domain spasial. Sedangkan pada filter spasial, pada citra dilakukan konvolusi dengan sebuah *kernel* [21]

2.6.1 Mean Filter

Mean filter digunakan untuk *smoothing* citra, mereduksi variasi intensitas piksel antara satu piksel dengan yang lain. Filter ini banyak digunakan untuk mereduksi *noise* pada citra. Inti dari *mean filtering* adalah mengganti setiap nilai piksel pada citra dengan nilai rata-rata piksel tetangganya [22]. *Kernel* dari *mean filter* ditunjukkan pada **Gambar 2.12**

Cara kerja *mean filter* yang mengganti nilai piksel dengan rata-rata nilai tetangganya mengakibatkan hilangnya nilai piksel yang terlalu rendah atau tinggi dibanding tetangganya. Piksel yang terlalu rendah atau tinggi ini bisa jadi merupakan sebuah *noise* karena nilainya yang jauh berbeda dengan piksel sekitarnya.

2.6.2 Median Filter

Median filter digunakan untuk menurunkan *noise* pada citra seperti halnya *mean filter*, namun lebih baik menjaga detail pada citra. Inti dari *median filter* adalah mengganti setiap nilai piksel pada citra dengan median dari nilai piksel-piksel tetangganya [23]. Contoh dari *median filter* ditunjukkan pada **Gambar 2.13**

Median filter lebih *robust* daripada *mean filter* karena dengan menggunakan perhitungan median, maka nilai piksel yang terlalu tinggi atau rendah tidak akan mempengaruhi nilai median, keterbalikan dari *mean filter* yang menggunakan perhitungan rata-rata. Karena nilai yang dihasilkan juga adalah nilai piksel yang ada di tetangga piksel itu sendiri, bukan nilai baru seperti pada *mean filter*, maka *median filter* juga lebih baik dalam menjaga ketajaman tepi citra.

2.6.3 Motion Blur Filter

Motion blur adalah efek tertangkapnya benda bergerak dalam suatu citra atau foto diam. *Motion blur* pada pengolahan citra digital digunakan untuk memberikan efek seperti gerakan bergeser pada citra atau foto. *Motion blur* dapat menurunkan intensitas piksel yang arahnya berlawanan dengan arah gerakan pada *blur*.

Inti dari *motion blur filter* adalah mengkonvolusi citra dengan sebuah *structuring element* dengan sudut yang ditentukan [23]. Hasil konvolusi akan mengakibatkan efek seperti gerakan bergeser pada citra. *Kernel* dari *motion blur* ditunjukkan pada **Gambar 2.14**, sedangkan contoh hasil *motion blur* ditunjukkan pada **Gambar 2.15**

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Gambar 2.12 *Kernel mean filter*

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

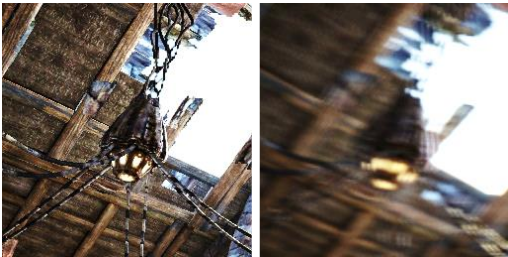
Neighbourhood values:
115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

Gambar 2.13 *Median filtering*

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

Gambar 2.14 *Kernel motion blur*

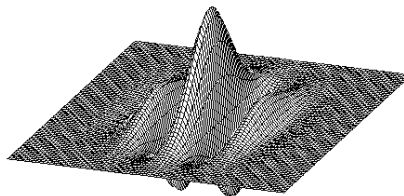


Gambar 2.15 Hasil *Motion Blur*

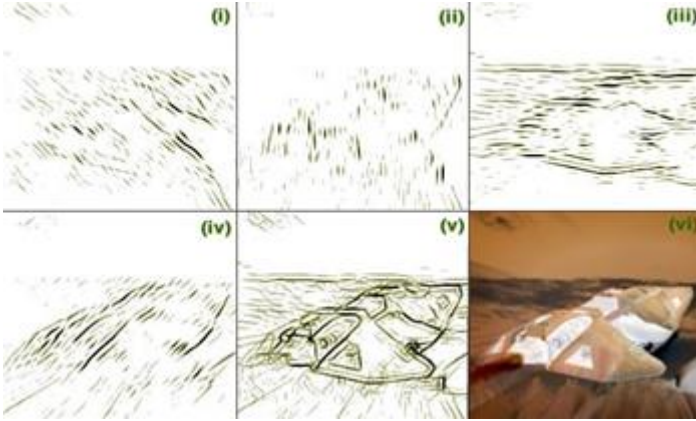
2.6.4 Gabor Filter

Gabor filter adalah filter *bandpass* yang digunakan pada pengolahan citra untuk ekstraksi fitur, analisis tekstur dan *stereo disparity estimation* [24]. Filter gabor didapatkan dengan modulasi sinyal *sinusoid* dengan sinyal *gaussian*. Filter tersebut memproses sinyal masukan pada bagian frekuensi tertentu. Pada sinyal berdimensi satu, sinyal *sinusoid* dari dimensi dengan frekuensi tertentu dimodulasikan dengan sinyal *gaussian*. Pada sinyal berdimensi dua, sinyal *sinusoid* yang digunakan, memiliki frekuensi dan arah sehingga filter gabor yang dihasilkan juga memiliki arah. Bentuk dari filter gabor ditunjukkan pada **Gambar 2.16**

Satu filter gabor merepresentasikan satu orientasi saja, sehingga dibutuhkan beberapa operasi *filtering* untuk memproses citra masukan pada beberapa orientasi yang berbeda. Satu filter gabor juga mengandung satu frekuensi saja yang merepresentasikan ukuran filter. Pada citra filter gabor direpresentasikan berupa sebuah *kernel* atau *array* dua dimensi yang digunakan untuk operasi konvolusi pada citra masukan. Operasi konvolusi dengan *kernel* tersebut menghasilkan deteksi tepi atau garis yang searah dengan *kernel*. Garis-garis pada citra yang tidak searah dengan *kernel* akan melewati bagian lembah dari *kernel* dan menghasilkan nilai yang kecil, garis-garis tersebut tidak masuk dalam citra hasil konvolusi. Hasil dari konvolusi tiap filter dengan citra kemudian disatukan untuk mendapat citra keluaran. Hasil filter gabor ditunjukkan pada **Gambar 2.17** Gambar tersebut diinvers warnanya agar lebih mudah dibaca.



Gambar 2.16 Gabor Filter



Gambar 2.17 Hasil filter gabor pada sudut (i) 135 derajat, (ii) 90 derajat, (iii) 0 derajat, (iv) 45 derajat, kemudian disatukan untuk menghasilkan (v) output filter gabor dari (vi) gambar asli

Filter Gabor dengan frekuensi θ dan orientasi ϕ dituliskan pada persamaan (2.3) berikut:

$$g(x, y, \theta, \phi) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \exp(2\pi\phi i(x \cos \phi + y \sin \phi)) \quad (2.3)$$

Parameter yang menentukan bentuk dan ukuran dari filter Gabor yaitu: panjang gelombang atau *wavelength* (λ) yang menentukan panjang gelombang *kernel* filter Gabor, orientasi (θ) yang menentukan arah perputaran filter Gabor untuk mendeteksi struktur dengan arah tertentu, *phase offset* yang menentukan pergeseran sinyal kosinus, *aspect ratio* (γ) yang menentukan elipsitas bentuk filter Gabor dan *bandwith* (b) yang menentukan berapa fase kosinus yang tercakup dalam filter Gabor.

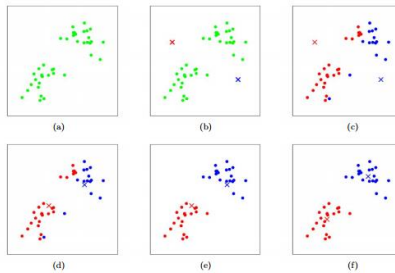
2.7 K-means Clustering

K-means clustering adalah salah satu metode *unsupervised learning*, yaitu algoritma *machine learning* yang bertujuan untuk membentuk kelas-kelas atau kluster pada data yang belum atau tidak diketahui kelasnya [25] *K-means clustering* adalah metode *clustering* untuk mempartisi sebuah data menjadi K kluster.

K-means clustering mengambil beberapa titik acak c_k sebagai titik pusat tiap kluster. Tiap titik pada data lalu dievaluasi jaraknya dengan masing-masing titik pada c_k hingga dapat ditemukan tiap titik adalah anggota kluster apa. Dalam perhitungan matematis J adalah jarak antara titik $x_i^{(j)}$ ke titik pusat c_j [26]. Nilai J dapat dihitung dengan formula (2.4) berikut:

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2 \quad (2.4)$$

Langkah selanjutnya, data yang sudah dikluster dilakukan penghitungan kembali titik pusat kluster untuk mendapat titik pusat baru menggantikan c_k . Titik pusat baru didapatkan dari rata-rata atribut seluruh anggota kluster. Setelah titik pusat baru didapat, dilakukan perulangan kembali menghitung jarak pada tiap titik ke titik-titik pusat kluster yang baru. *K-means clustering* berhenti ketika titik pusat tiap kluster tidak berubah nilainya ketika dilakukan penghitungan kembali titik pusat kluster. Proses *K-means clustering* diilustrasikan pada **Gambar 2.18**



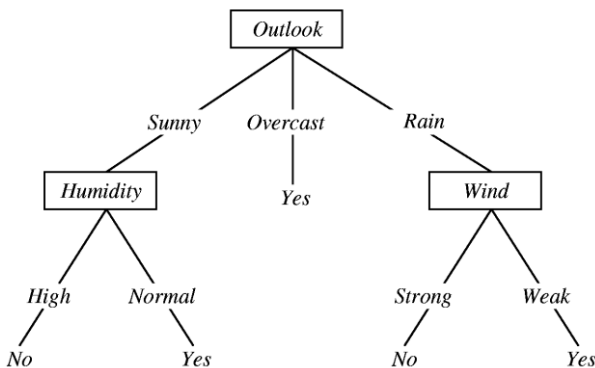
Gambar 2.18 Ilustrasi *K-means Clustering*

2.8 Decision Tree

Decision Tree adalah salah satu metode *supervised learning*, yaitu algoritma *machine learning* yang bertujuan untuk membentuk suatu *rule* penentuan kelas dari data yang telah diketahui kelasnya, sehingga pada observasi data yang lain, data tersebut dapat diklasifikasi kepada kelas tertentu [25].

Decision tree adalah prosedur klasifikasi yang dilakukan untuk membentuk aturan partisi dari sampel data secara rekursif. Data sampel dikelompokkan menjadi beberapa *box*, dan pada tiap langkah prosedur, akan dicek apakah data dalam *box* tersebut dapat dipartisi menjadi dua *box* baru [25]. Partisi data tersebut ditentukan oleh sebuah kondisi.

Decision tree mempunyai struktur berupa graf berbentuk *tree* yang berisi kumpulan kondisi atau hasil pada titik yang disebut *node*. *Node* awal dari *decision tree* dimana klasifikasi bermula disebut sebagai *root*. Tiap *node* dapat memiliki atau tidak memiliki sebuah kondisi yang dapat bersifat diskrit maupun kontinu. Kondisi digunakan untuk mengetes atribut dari data yang diklasifikasi. *Node* yang memiliki kondisi mempunyai cabang (*branch*) menuju *node* lainnya. *Node* yang tidak memiliki kondisi disebut *leaf*. *Leaf* berisi keputusan nilai label dari data yang diklasifikasi. Struktur *decision tree* ditunjukkan pada **Gambar 2.19**



Gambar 2.19 Struktur *decision tree*

Pada *decision tree* dapat dilakukan teknik *pruning*, yaitu pengurangan ukuran *decision tree*. Pengurangan ukuran ini dilakukan dengan cara menghilangkan bagian dari *tree* yang memberikan efek kecil dalam klasifikasi. *Pruning* dapat membantu meningkatkan akurasi prediksi dari *decision tree* dan menghindari *overfitting*. *Overfitting* sendiri adalah keadaan dimana sebuah *classifier* gagal membentuk *rule* klasifikasi yang baik karena kekompleksan data.

2.8.1 Algoritma C4.5

Salah satu metode untuk membangun *decision tree* adalah algoritma C4.5 yang dikembangkan oleh Quinlan [27]. Pada algoritma ini diketahui sebuah data yang mempunyai atribut dan label kelas. Pada algoritma ini tiap data dan tiap atribut dicek apakah data dapat dipartisi menjadi bagian yang lebih kecil. Untuk menentukan lokasi partisi, dihitung sebuah nilai statistik untuk mengetahui persebaran kelas jika dilakukan partisi di lokasi tersebut (entropi). Secara matematis dapat dilambangkan S sebagai data, p_+ sebagai proporsi kelas positif dan p_- sebagai proporsi kelas negatif. Rumus entropi ditunjukkan pada persamaan (2.5) berikut.

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \quad (2.5)$$

Dimana Nilai entropi digunakan sebagai dasar menghitung *information gain*. *Information gain* merepresentasikan perbedaan jumlah informasi yang dibutuhkan untuk mengidentifikasi element T sebelum dan setelah partisi di lokasi tersebut. Secara matematis dapat dilambangkan A sebagai atribut, dan S_v adalah subset S dimana atribut A mempunyai nilai v . Rumus *information gain* ditunjukkan pada persamaan (2.6) berikut.

$$Gain(S, A) = Entropy(S) - \sum_{v=values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2.6)$$

Nilai *information gain* dapat digunakan untuk meranking kepentingan suatu lokasi partisi. Lokasi partisi dengan nilai *information gain* terbesar adalah lokasi partisi yang dipilih menjadi kondisi dalam *node*.

2.8.2 Root-Guided Decision Tree

Root-Guided Decision Tree (RGDT) adalah variasi algoritma pembangunan *decision tree* C4.5. Perbedaan RGDT dengan C4.5 adalah, pada RGDT dibentuk sekumpulan *decision tree* pada awal mula pembuatan *rule*. Setiap atribut pada data menjadi *root* dari masing-masing *decision tree* yang dibentuk. Setelah setiap atribut ditentukan menjadi *root* pada setiap *tree*, pembentukan keseluruhan *tree* dilakukan seperti algoritma C4.5. Percabangan ditentukan dari nilai *gain* terbesar dan dilakukan sampai sebelum terjadi *overfitting*. *Decision tree* dengan hasil akurasi tertinggi dari kumpulan *decision tree* yang dimiliki kemudian dijadikan dasar untuk membuat *rule* klasifikasi [8].

Contoh kasus untuk RGDT adalah pada klasifikasi spesies hewan. Misalnya, terdapat data hewan dengan empat atribut yaitu cara bernapas, cara melahirkan, makanan dan habitat. Hewan dalam data ingin diklasifikasikan sebagai mamalia dan nonmamalia. Dalam RGDT, akan dibentuk *decision tree* sebanyak jumlah atribut pada data, yaitu empat. Setiap atribut menjadi *root* dari setiap *decision tree* yang dibentuk. Kemudian, setiap *tree* disusun hingga lengkap seperti algoritma C4.5. Dari setiap *decision tree* yang terbentuk, dihitung nilai akurasinya. Didapat *decision tree* dengan atribut cara melahirkan sebagai *root* mampu mengklasifikasikan data hewan sebagai mamalia dan nonmamalia dengan akurasi tertinggi. Oleh karena itu, *decision tree* dengan atribut cara melahirkan dipakai sebagai dasar pembuatan *rule* klasifikasi seluruh data.

2.9 Confusion Matrix

Confusion matrix adalah sebuah statistik klasifikasi yang menyimpan informasi mengenai prediksi kelas dan kelas asli. *Confusion matrix* banyak digunakan untuk menguji performa dari

suatu metode klasifikasi [28]. Pada tugas akhir ini, *confusion matrix* mencatat informasi klasifikasi piksel menjadi kelas area *trabecular bone* atau area *non-trabecular bone*. Struktur *confusion matrix* ditunjukkan pada **Gambar 2.20**

Dalam *confusion matrix* pada tugas akhir ini terdapat statistik *true positive* (TP) yaitu piksel area *trabecular bone* yang terklasifikasi sebagai area *trabecular bone*, *true negative* (TN) yaitu piksel area *non-trabecular bone* yang terklasifikasi sebagai area *non-trabecular bone*, *false positive* (FP) yaitu piksel area *non-trabecular bone* yang terklasifikasi sebagai area *trabecular bone*, *false negative* (FN) yaitu piksel area *trabecular bone* yang terklasifikasi sebagai area *non-trabecular bone*

		Sistem	
Aktual		Trabecular bone	Non-trabecular bone
	Trabecular bone	TP	FN
	Non-trabecular bone	FP	TN

Gambar 2.20 *Confusion Matrix* berdasar Area

Dari *confusion matrix* bisa didapatkan berbagai informasi mengenai performa *classifier*, di antaranya akurasi, *specificity*, dan *sensitivity*. Akurasi adalah jumlah total data yang prediksi kelas hasil klasifikasinya sesuai dengan data *ground truth* [28]. Rumus perhitungan dari akurasi ditunjukkan pada persamaan (2.7)

$$Akurasi = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.7)$$

Specificity adalah jumlah proporsi data yang prediksi kelas hasil klasifikasinya adalah salah dari keseluruhan data yang berkelas salah pada *ground truth*. Rumus perhitungan *specificity* ditunjukkan pada persamaan (2.8)

$$Specificity = \frac{TN}{FP+TN} \quad (2.8)$$

Sensitivity atau *recall* adalah jumlah proporsi data yang prediksi kelas hasil klasifikasinya adalah benar dari keseluruhan data yang berkelas benar pada *ground truth*. Rumus perhitungan *specificity* ditunjukkan pada persamaan (2.9)

$$Sensitivity = \frac{TP}{TP+FN} \quad (2.9)$$

Precision adalah jumlah data yang pada *ground truth* benar dari keseluruhan data yang terklasifikasi benar. Rumus perhitungan *specificity* ditunjukkan pada persamaan (2.10)

$$Precision = \frac{TP}{TP+FP} \quad (2.10)$$

F1 Score atau *F-measure* adalah pengukuran akurasi dengan memperhatikan nilai *precision* dan *recall* untuk mengetahui kinerja *classifier* dalam mendapatkan informasi yang diinginkan. Rumus perhitungan *F1 Score* ditunjukkan pada persamaan (2.11)

$$F1\ Score = \frac{2TP}{2TP+FP+FN} \quad (2.11)$$

BAB III

ANALISIS DAN PERANCANGAN

Bab analisis dan perancangan berisi analisis kebutuhan dan perancangan aplikasi yang akan dibangun. Tahap analisis membahas mengenai analisis kebutuhan yang menjadi dasar dari tahap perancangan.

3.1 Tahap Analisis

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan aplikasi segmentasi area *trabecular bone*. Selain itu dijelaskan pula alasan pengerjaan masing-masing tahap pada tugas akhir ini.

3.1.1 Deskripsi Umum

Pada tugas akhir ini dibangun aplikasi untuk melakukan segmentasi area *trabecular bone* pada *dental panoramic radiograph* (DPR). Data masukan yang digunakan adalah data *scan* DPR. Data keluaran dari aplikasi merupakan citra biner (hitam putih) yang merepresentasikan hasil segmentasi area *trabecular bone*.

Aplikasi ini diharapkan dapat digunakan untuk menunjang pemanfaatan DPR di bidang kesehatan. Hasil segmentasi area *trabecular bone* yang akurat dapat dipergunakan sebagai dasar dalam perhitungan informasi struktur *trabecular bone* pada DPR. Hasil perhitungan informasi struktur *trabecular bone* ini dapat digunakan untuk mengidentifikasi berbagai penyakit, contohnya osteoporosis.

3.1.2 Spesifikasi Kebutuhan Sistem

Pada aplikasi segmentasi area *trabecular bone* dibutuhkan beberapa proses untuk dapat memenuhi kebutuhan sistem dalam menghasilkan segmentasi yang akurat. Proses tersebut antara lain:

1. *Preprocessing*

Preprocessing dilakukan untuk memperbaiki citra masukan berupa *Region of Interest* (ROI) dari citra asli.

2. Ekstraksi fitur

Ekstraksi fitur dilakukan untuk mendapatkan data yang representatif terhadap bentuk objek dalam citra.

3. *Clustering*

Clustering dilakukan untuk mengelompokkan piksel menjadi kluster-kluster sesuai objek yang ada pada ROI.

4. Klasifikasi

Klasifikasi dilakukan untuk memperbaiki kesalahan klasifikasi piksel dalam proses *clustering*.

5. *Postprocessing*

Postprocessing dilakukan untuk memperbaiki citra yang dihasilkan pada proses klasifikasi.

3.1.3 Analisis Permasalahan

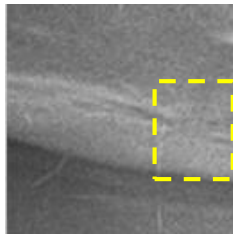
Hasil segmentasi area *trabecular bone* dapat dijadikan dasar dalam perhitungan informasi struktur *trabecular bone*. Informasi struktur *trabecular bone* yang dapat dihitung diantaranya percabangan, luas rongga antartulang dan sebagainya yang berguna dalam identifikasi penyakit, contohnya osteoporosis. Hasil segmentasi yang akurat akan memberikan hasil perhitungan informasi struktur *trabecular bone* yang lebih akurat pula.

Untuk mencapai hasil segmentasi yang akurat, terdapat beberapa permasalahan yang dapat menurunkan akurasi segmentasi. Permasalahan-permasalahan tersebut dijelaskan sebagai berikut:

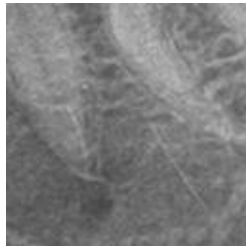
3.1.3.1 Analisis Permasalahan *Preprocessing*

Pada ROI, terdapat perbedaan kontras antara ROI bagian atas, yang berisikan area *trabecular bone* dan akar gigi, dan ROI bagian bawah, yang berisikan area *trabecular bone*, *cortical bone* dan area di bawah rahang. Perbedaan kontras terutama terlihat di antara area *cortical bone* dan *trabecular bone* seperti ditunjukkan pada **Gambar 3.1**. Hal ini mengakibatkan segmentasi yang dilakukan secara langsung pada ROI memberikan hasil yang kurang akurat. Oleh karena itu, diperlukan pemisahan ROI menjadi dua bagian, yaitu ROI atas dan ROI bawah.

Detail struktur *trabecular bone* yang melintang di area *trabecular bone* seperti ditunjukkan pada **Gambar 3.2** juga mengakibatkan terganggunya proses segmentasi karena intensitas piksel pada struktur *trabecular bone* keterangan warnanya mirip dengan struktur gigi atau *cortical bone*. Jika tidak diperhalus atau dihilangkan, akan mengakibatkan algoritma *clustering* dan klasifikasi menganggap struktur *trabecular bone* tersebut sebagai *cortical bone* atau gigi. Oleh karena itu perlu dilakukan penghilangan struktur *trabecular bone* tersebut menggunakan morfologi *opening* dan *motion blur*. Adanya *noise* dalam citra dihilangkan menggunakan *median filter*. Selain itu, untuk menyamaratakan rentang intensitas piksel untuk seluruh citra ROI pada data, dilakukan normalisasi rentang baru pada rentang 0-255.



Gambar 3.1 ROI bawah. Area dalam kotak kuning memperlihatkan rendahnya kontras

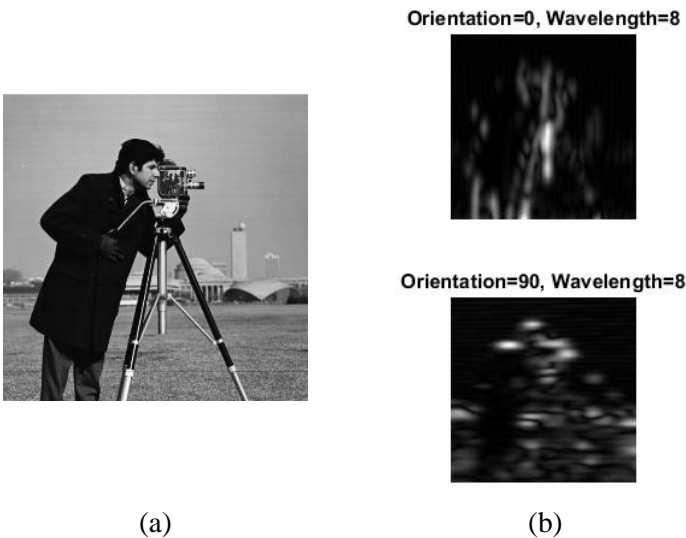


Gambar 3.2 Detail Struktur Trabecular Bone yang terdapat diantara dua akar gigi

3.1.3.2 Analisis Permasalahan Ekstraksi Fitur

Ekstraksi fitur perlu dilakukan dengan baik, karena data fitur yang sesuai dan representatif terhadap bentuk objek dalam ROI

akan menghasilkan hasil segmentasi yang akurat. Dalam tugas akhir ini, ekstraksi fitur dilakukan dengan menggunakan filter Gabor. Filter Gabor adalah piksel yang memberikan nilai tinggi pada objek pada citra yang orientasinya searah dengan orientasi filter Gabor, dan sebaliknya memberikan nilai yang rendah pada objek pada citra yang orientasinya berlawanan arah dengan orientasi filter Gabor. Pengaruh orientasi terhadap keluaran filter gabor ditunjukkan pada **Gambar 3.3**. Filter Gabor sendiri memiliki beberapa parameter yang dapat diatur untuk mendapatkan keluaran yang sesuai kebutuhan. Oleh karena itu dengan ekstraksi fitur oleh filter Gabor yang mempertimbangkan orientasi objek pada citra ROI, dapat memberikan data fitur yang representatif terhadap objek pada ROI.



Gambar 3.3 Pengaruh orientasi pada filter Gabor
(a) gambar asli (b) keluaran filter Gabor

Selain itu dilakukan pula ekstraksi fitur informasi spasial. Informasi spasial penting untuk dimasukkan dalam data fitur karena pada citra, satu piksel berkaitan dengan piksel-piksel

tetangganya. Selain itu, informasi spasial pada citra dapat menghindari timbulnya *noise*.

3.1.4 Analisis Permasalahan *Clustering*

Clustering dilakukan untuk mengelompokkan piksel menjadi dua kelompok yakni area *trabecular bone* dan area *nontrabecular bone* menggunakan metode *K-means clustering*. Pada ROI atas, ROI dikelompokkan menjadi kelas area akar gigi dan area *trabecular bone*. Pada ROI bawah, ROI dikelompokkan menjadi tiga *cluster* yaitu kelas area *trabecular bone*, kelas area *cortical bone*, dan kelas area nonrahang. Hal ini dilakukan karena pada ROI bawah area *cortical bone* dan area *trabecular bone* memiliki selisih internstias piksel yang lebih kecil daripada dibandingkan dengan area nonrahang sehingga apabila dikelompokkan menjadi dua *cluster*, maka area *trabecular bone* dan area *cortical bone* akan tercampur menjadi satu *cluster*.

Kelemahan dari metode *K-means clustering* adalah ukuran tiap *cluster* yang dihasilkan selalu sama. Hal ini mengakibatkan ambiguitas keanggotaan *cluster* terhadap piksel yang jaraknya jauh dari titik pusat *clusternya*. Dalam menentukan suatu piksel sebagai piksel yang jauh dari titik pusat *clusternya*, dilakukan pengukuran jarak dengan menghitung *eucledian distance* antara data nilai fitur suatu piksel dengan data nilai fitur titik pusat *clusternya*. Ditentukan sebuah nilai *threshold* untuk mengkategorikan suatu piksel sebagai piksel yang jaraknya jauh dari titik pusat *clusternya*. Piksel yang jaraknya dengan titik pusat *cluster* berada di atas nilai *threshold* akan diklasifikasi ulang dengan menggunakan RGDT

3.1.5 Analisis Permasalahan Klasifikasi

Klasifikasi ulang dilakukan pada piksel yang jaraknya ke titik pusat *clusternya* di atas nilai *threshold*. Proses klasifikasi ulang ini memerlukan *classifier* yang mempunyai nilai akurasi yang tinggi sehingga tidak malah memperburuk hasil segmentasi yang didapat dari *K-means clustering*. Pada tugas akhir ini digunakan algoritma RGDT dalam penyusunan *decision tree*. RGDT membangun kumpulan *tree* dengan masing-masing atribut sebagai *node*. Kemudian dipilih *tree* dengan nilai akurasi terbaik,

sehingga hasil *decision tree* yang didapat merupakan *decision tree* dengan akurasi tertinggi.

Pada tugas akhir ini piksel tidak dilakukan klasifikasi langsung dengan *decision tree* karena *decision tree* mengklasifikasi piksel secara individual berdasarkan nilai fitur piksel itu sendiri, sehingga dapat menimbulkan *noise* pada citra hasil segmentasi. Sedangkan pada *K-means clustering* dalam penentuan *cluster* satu piksel masih berkaitan dengan piksel lainnya lewat posisi *centroid* yang selalu diupdate setiap iterasi berdasar rata-rata nilai piksel pada *cluster*. Oleh karena itu, pada tugas akhir ini dilakukan *K-means clustering* terlebih dahulu kemudian hasil *K-means clustering* dievaluasi ulang oleh RGDT.

Setelah proses klasifikasi selesai, pada ROI bawah dilakukan penggabungan kelas area *cortical bone* dan area nonrahang menjadi satu kelas area non-*trabecular bone*, sehingga pada akhir proses klasifikasi, pada ROI bawah terdapat dua *cluster* yaitu kelas area *trabecular bone* dan area non-*trabecular bone*.

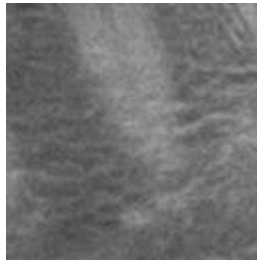
3.1.6 Analisis Permasalahan *Postprocessing*

Postprocessing dilakukan untuk memperbaiki kualitas citra yang dihasilkan pada proses klasifikasi. *Postprocessing* dilakukan karena pada area *trabecular bone* terdapat area-area terang yang intensitas pikselnya mirip dengan objek non-*trabecular bone* seperti akar gigi atau *cortical bone*. Hal ini ditunjukkan pada **Gambar 3.4**

Area-area terang tersebut mempunyai luas yang lebih kecil apabila dibandingkan dengan area objek akar gigi maupun *cortical bone*. Hal ini dapat dijadikan dasar untuk menghapus area-area terang tersebut dengan cara membandingkan luas area-area terang tersebut dengan luas area objek non-*trabecular bone*. Area-area terang yang mempunyai luas kecil tersebut kemudian dapat dihapus.

3.2 Tahap Perancangan

Tahap perancangan dilakukan untuk merancang proses secara keseluruhan berdasarkan fungsionalitas dan kebutuhan dari aplikasi segmentasi area *trabecular bone*.



Gambar 3.4 Contoh Area-Area Terang pada ROI

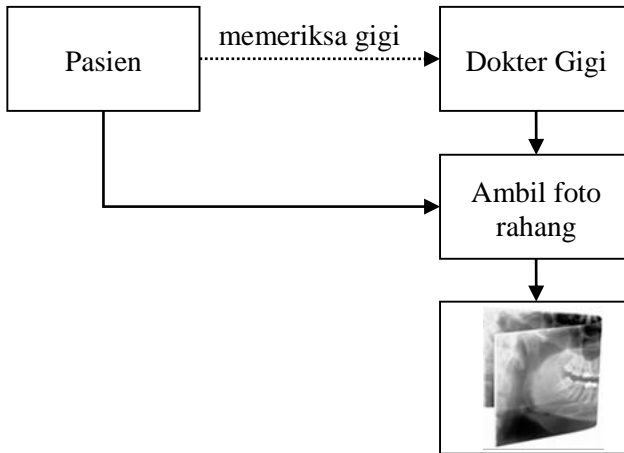
3.2.1 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi segmentasi area *trabecular bone*. Untuk dapat menggunakan aplikasi segmentasi area *trabecular bone*, pengguna harus memiliki DPR sebagai data masukan. Data masukan diperoleh dari foto *x-ray* rahang pasien dokter gigi. Ilustrasi pengambilan foto rahang ditunjukkan pada **Gambar 3.5**.

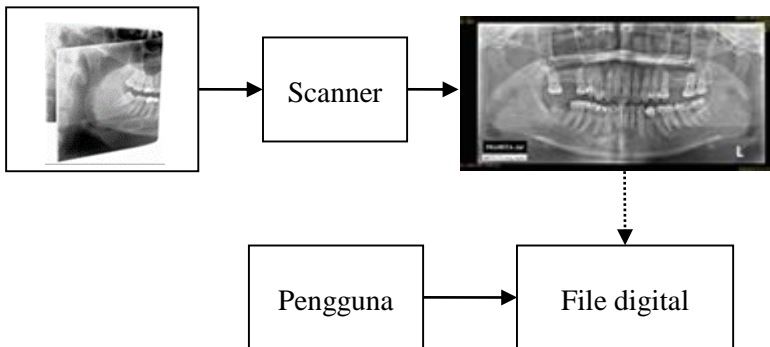
Jika mesin *x-ray* yang digunakan masih menggunakan teknologi film, *file* gambar tidak bisa disimpan dalam format digital. Oleh karena itu, perlu dilakukan *scanning* gambar terlebih dahulu untuk mendapatkan DPR dalam format digital. Ilustrasi proses ditunjukkan pada **Gambar 3.6**

Sistem segmentasi pada tugas akhir ini memiliki tiga proses utama. Proses pertama adalah ekstraksi fitur. Proses kedua adalah *clustering*. Proses ketiga adalah klasifikasi. Proses keempat adalah *postprocessing*. Diagram alir dari sistem ditunjukkan pada **Gambar 3.7**

Tahap *preprocessing* dilakukan terlebih dahulu sebelum memasuki proses utama aplikasi. Tahap ini terdiri dari pembagian ROI menjadi bagian atas dan bawah normalisasi rentang keabuan pada citra dan pengaburan detail struktur *trabecular bone* pada citra.



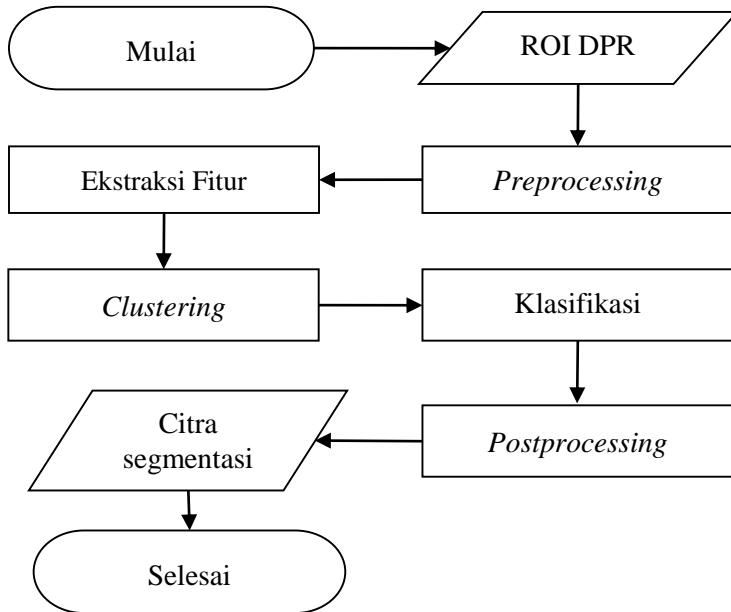
Gambar 3.5 Ilustrasi Pengambilan Foto Rahang



Gambar 3.6 Ilustrasi Proses *Scanning* Foto

Ekstraksi fitur adalah proses untuk mendapatkan vektor fitur pada tiap piksel citra yang telah melalui proses *preprocessing*. Nilai dari vektor fitur tersebut digunakan dalam proses *clustering* dan pembangunan RGDT pada tahap *training* data. Tugas akhir ini

menggunakan fitur berupa: keluaran *Gabor filter* dari ROI bagian bawah dan ROI bagian atas, informasi spasial nilai intensitas piksel dan koordinat piksel di sumbu y khusus untuk ROI bagian bawah.



Gambar 3.7 Diagram Alir dari Sistem Aplikasi Segmentasi area Trabecular Bone

Proses *clustering* adalah proses untuk mengelompokkan piksel-piksel pada citra sebagai kelas area *trabecular bone* dan kelas area *non-trabecular bone*. *Clustering* dilakukan berdasarkan data fitur yang didapat pada proses ekstraksi fitur. Setelah terbagi menjadi kelas-kelas, dilakukan *thresholding* untuk mendapatkan piksel-piksel terjauh dari pusat *cluster* yang akan diklasifikasi ulang oleh RGDT.

Proses klasifikasi adalah proses untuk mengklasifikasi ulang piksel-piksel terjauh dari pusat *cluster* menggunakan RGDT. Proses klasifikasi didahului dengan penyusunan *decision tree* menggunakan *training* data. Atribut pada data *training* adalah

atribut hasil ekstraksi fitur tanpa atribut informasi spasial dan koordinat sumbu y, dan ditambahkan atribut label kelas dari *ground truth*. Selanjutnya dilakukan proses *testing*, yaitu pengelompokkan piksel menggunakan *decision tree* dengan memanfaatkan vektor fitur dari proses ekstraksi fitur. Pada akhir klasifikasi, kelas area *cortical bone* dan kelas area nonrahang digabung menjadi satu kelas area non-*trabecular bone*.

Proses *postprocessing* adalah proses untuk memperbaiki citra segmentasi yang dihasilkan oleh proses klasifikasi. Pada tahap ini, dihitung area dari setiap *region* yang ada dalam citra segmentasi. Apabila ukurannya relatif kecil dibandingkan objek lainnya, maka *region* tersebut dianggap sebagai *noise* dan dihapus. Pada tahap ini dihasilkan citra biner yang merepresentasikan hasil segmentasi area *trabecular bone*.

3.2.2 Perancangan Data

Perancangan data dilakukan untuk memastikan pengoperasian aplikasi berjalan dengan benar. Data masukan (*input*) adalah data yang diperlukan dalam pengoperasian aplikasi dan data keluaran (*output*) adalah data yang akan digunakan oleh pengguna.

Data masukan adalah data yang diproses oleh aplikasi segmentasi area *trabecular bone*. Data yang digunakan sebagai masukan adalah DPR. Data masukan berupa potongan ROI berukuran 256x128 piksel dengan citra abu (*grayscale*). Contoh citra yang digunakan sebagai data masukan ditunjukkan pada **Gambar 3.8**

Data keluaran aplikasi segmentasi area *trabecular bone* merupakan citra biner hasil segmentasi dari data masukan. Bagian yang berwarna hitam pada citra merupakan area *trabecular bone*, sedangkan bagian yang berwarna putih merupakan area non-*trabecular bone*. Contoh citra keluaran ditunjukkan pada **Gambar 3.9**

3.2.3 Perancangan Proses

Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada aplikasi segmentasi area

trabecular bone. Bagan dari setiap proses utama aplikasi dapat dilihat pada **Gambar 3.7**.

3.2.3.1 Preprocessing

Supaya aplikasi dapat berjalan optimal, diperlukan persiapan data terlebih dahulu pada tahap *preprocessing*. Pada tugas akhir kali ini, tahap *preprocessing* yang dilakukan adalah membagi ROI menjadi bagian atas dan bawah, normalisasi citra, *median filter*, morfologi dengan *structuring element* serta *motion blur*. Diagram alir dari proses ditunjukkan pada **Gambar 3.10**.

Pembagian ROI menjadi bagian atas dan bawah dilakukan dengan cara memotong ROI sepanjang koordinat 128 pada sumbu Y. Bagian ROI pada koordinat 1-128 menjadi ROI bagian bawah dan bagian ROI pada koordinat 129-256 menjadi ROI bagian atas. Pemotongan ROI ditunjukkan pada **Gambar 3.11**

Normalisasi citra dilakukan untuk menyamakan rentang keabuan antarcitra pada data masukan. *Median filter* dilakukan untuk menghilangkan *noise* yang terdapat pada citra hasil normalisasi.

Structuring element dibuat untuk menghilangkan stuktur *trabecular bone*. Citra hasil normalisasi dimorfologi dengan *structuring element*. *Motion blur* dilakukan pada citra hasil morfologi dengan *structuring element* untuk memperhalus citra hasil morfologi tersebut. ROI, citra hasil morfologi dan citra hasil *motion blur* ditampilkan pada **Gambar 3.12**

3.2.3.2 Ekstraksi Fitur

Ekstraksi fitur dalam tugas akhir ini menggunakan data keluaran proses *Gabor filter*, informasi spasial dan koordinat sumbu Y. Hasil ekstraksi fitur kemudian digabungkan pada satu matriks Diagram alir dari proses ditunjukkan pada **Gambar 3.13**.

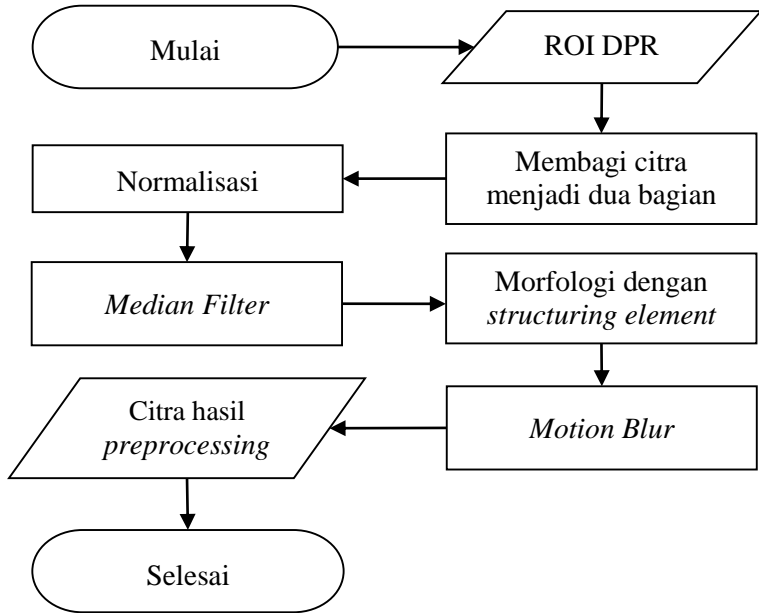
Gabor filter dilakukan untuk mendapatkan fitur berupa intensitas kekuatan garis dari tiap piksel pada ROI. *Gabor filter* dilakukan beberapa kali pada citra untuk mendapatkan beberapa atribut fitur. *Mean* dan *median filter* setelah *Gabor filter* digunakan untuk memperhalus citra dan menghilangkan *noise*. Hasil *Gabor filter* ditunjukkan pada **Gambar 3.14**



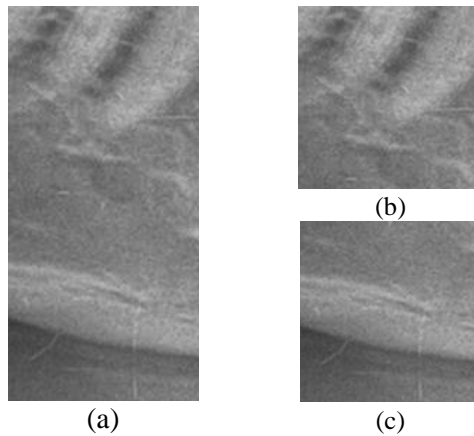
Gambar 3.8 Citra ROI



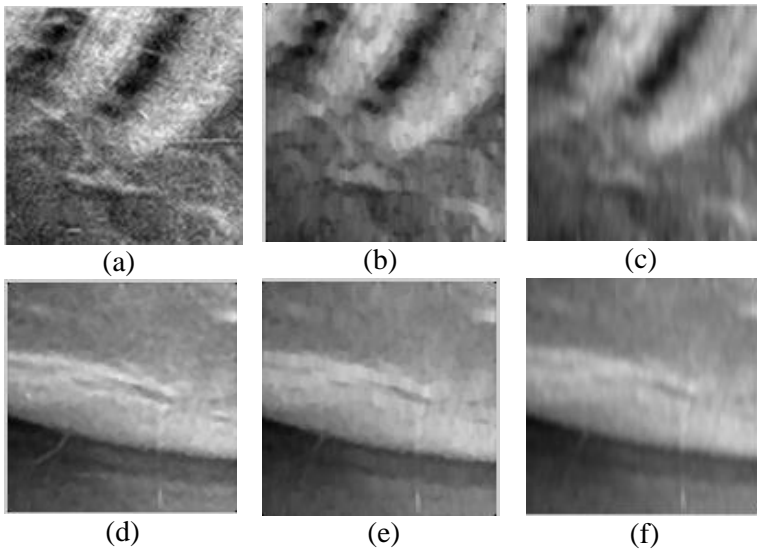
Gambar 3.9 Citra Hasil Segmentasi



Gambar 3.10 Diagram Alir Tahap *Preprocessing*



Gambar 3.11 Pemotongan ROI (a) ROI asli (b) ROI atas (c) ROI bawah

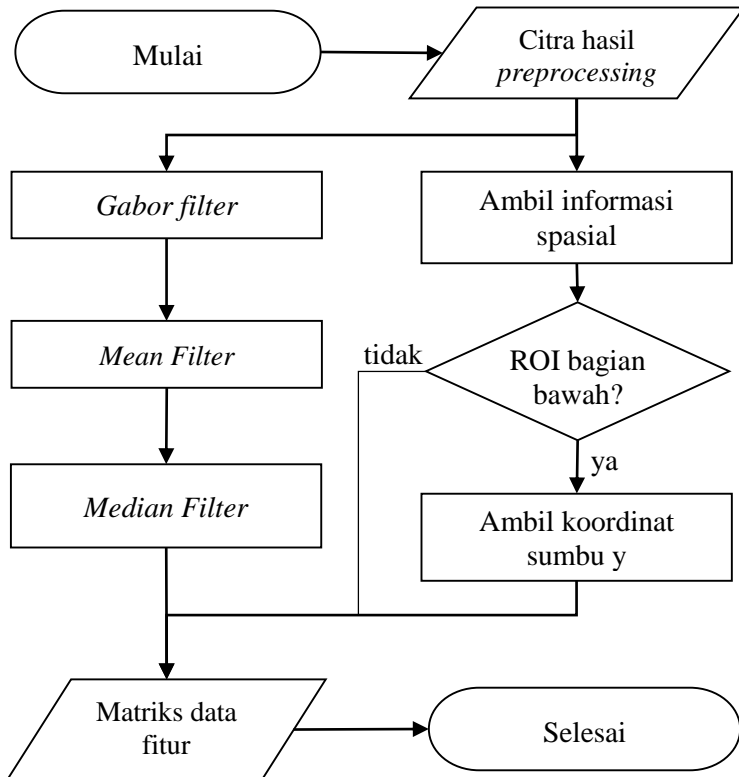


Gambar 3.12 Hasil *preprocessing*, (a) dan (d) normalisasi, (b) dan (e) morfologi *structuring element*, serta (c) dan (f) *motion blur*

.Pengambilan informasi spasial digunakan untuk mengetahui atribut nilai intensita piksel dari tetangga piksel tersebut. Informasi spasial didapatkan dengan menghitung rata-rata nilai piksel dari suatu *window* berukuran tertentu. Sebelum dihitung rata-ratanya, dilakukan replikasi matriks pada ROI.

Pengambilan informasi koordinat sumbu y dilakukan khusus pada ROI bagian bawah karena keanggotaan *cluster* suatu piksel sangat terkait dengan atribut koordinat sumbu y piksel tersebut.

Data fitur tiap atribut yang didapatkan dirubah dari matriks dua dimensi berukuran 128x128 menjadi matriks satu dimensi berukuran 16384x1. Kemudian tiap data fitur digabung menjadi satu matriks berukuran 16384x4 untuk ROI bagian atas dan 16384x5 untuk ROI bagian bawah.

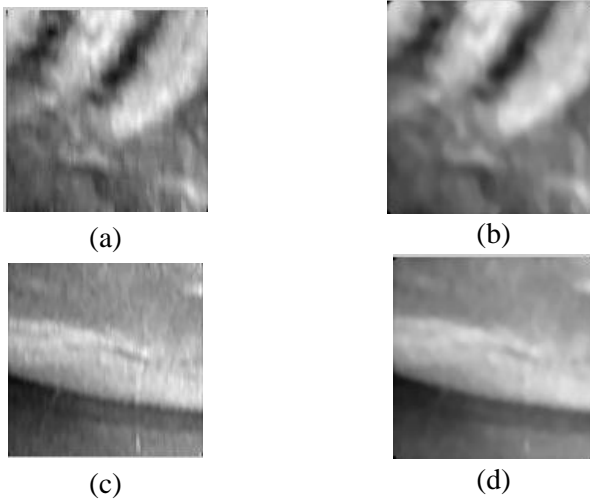


Gambar 3.13 Diagram Alir Proses Ekstraksi Fitur

3.2.3.3 Clustering

Proses *clustering* dilakukan untuk mengelompokkan piksel menggunakan *K-means clustering*. Data yang digunakan pada proses *clustering* adalah data yang dihasilkan dari proses ekstraksi fitur. Diagram alir dari proses ditunjukkan pada **Gambar 3.16**.

Pada ROI bagian atas, piksel di-*cluster* menjadi dua kelas (area gigi dan area *trabecular bone*) sedangkan untuk ROI bagian bawah, piksel di-*cluster* menjadi tiga kelas (area *trabecular bone*, area *cortical bone* dan area luar rahang). Kemudian dilakukan pemberian warna pada piksel sesuai dengan *cluster*-nya

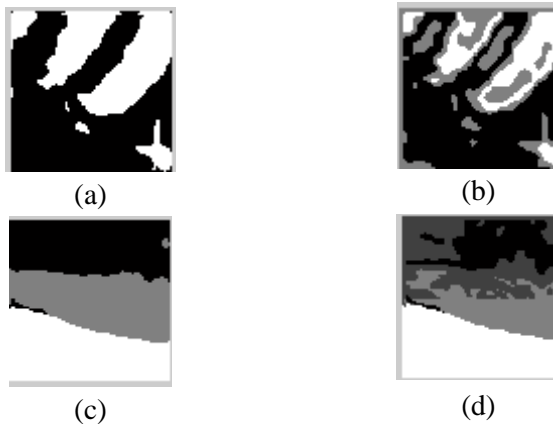


Gambar 3.14 Hasil *Gabor filter* (a) dan (c) tanpa *mean* dan *median filter* (b) dan (d) dengan *mean* dan *median filter*

Penghitungan jarak dari tiap piksel menuju titik pusat dari *cluster*-nya dilakukan dengan rumus *eucledian distance*. Piksel yang jaraknya dari titik pusat *cluster* di atas nilai *threshold* diklasifikasi ulang pada proses selanjutnya karena dianggap rawan salah klasifikasi, sedangkan yang jaraknya dari titik pusat *cluster* di bawah nilai *threshold* tidak diklasifikasi ulang karena dianggap sudah benar merupakan anggota *cluster* tersebut. Pada ROI Bawah piksel yang masuk *threshold* dibatasi yang koordinat sumbu Y piksel tersebut lebih kecil dari titik pusat *cluster area cortical bone*. Hasil *clustering* ditunjukkan pada **Gambar 3.15**

3.2.3.4 Klasifikasi

Proses klasifikasi *trabecular bone* adalah tahapan ketiga dari aplikasi segmentasi area *trabecular bone*. Diagram alir dari proses ditunjukkan pada **Gambar 3.18**. Pada proses ini, piksel yang jaraknya jauh dari titik pusat *cluster* pada proses sebelumnya diklasifikasi ulang. Penyusunan *decision tree* sebagai dasar pengklasifikasian ulang menggunakan metode RGDT.



Gambar 3.15 Hasil *Clustering* dan *Thresholding* pada (a) dan (b) ROI atas dan (c) dan (d) ROI bawah. Pixel di atas *threshold* diberi warna abu-abu untuk ROI atas dan abu-abu tua untuk ROI bawah

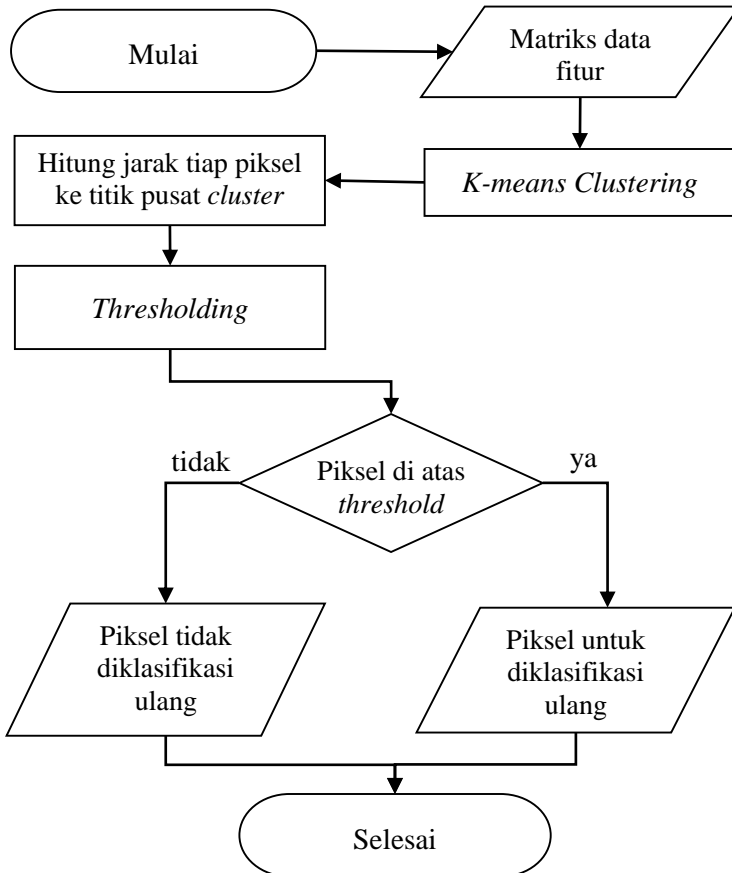
Langkah pertama pada proses ini adalah melakukan *training* dari *decision tree* menggunakan data fitur hasil dari proses ekstraksi fitur tanpa atribut informasi spasial dan atribut koordinat sumbu Y. Proses *training* data menghasilkan sebuah *struct* berisi *rule* dari *decision tree*. Kemudian, pixel yang diklasifikasi ulang di-*testing* dengan memasukkan atribut pixel tersebut pada *decision tree*. Kelas dari pixel diubah sesuai dengan hasil klasifikasi yang didapat

Pada akhir proses ini, pada ROI bawah dilakukan binerisasi pada ROI atas dan ROI bawah. Pada binerisasi ROI bawah dilakukan penggabungan kelas area *cortical bone* dan area nonrahang menjadi satu kelas area non-*trabecular bone*, sehingga pada akhir proses klasifikasi, pada ROI bawah terdapat dua *cluster* yaitu kelas area *trabecular bone* dan area non-*trabecular bone*. Hasil proses klasifikasi beserta binerisasinya ditunjukkan pada **Gambar 3.17**

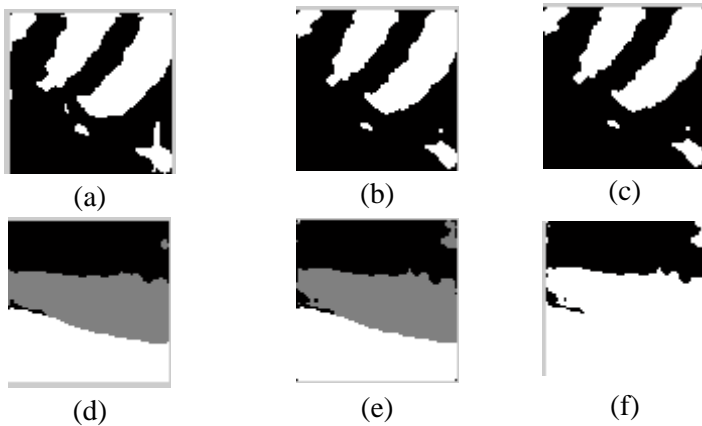
3.2.3.5 Postprocessing

Proses *postprocessing* adalah tahapan keempat dari aplikasi segmentasi area *trabecular bone*. Diagram alir dari proses

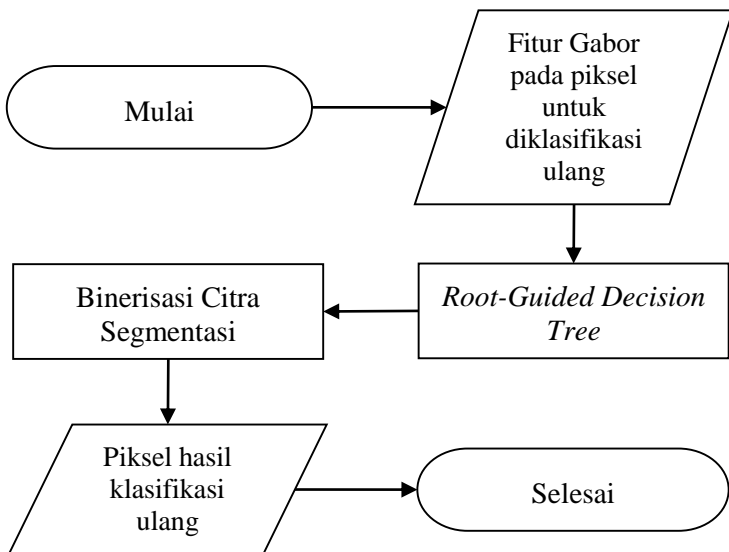
ditunjukkan pada **Gambar 3.20**. Pada proses ini, dilakukan penghapusan *noise* pada citra hasil klasifikasi. Pengukuran luas sebagai dasar pengklasifikasian ulang menggunakan fungsi *regionprops* MATLAB. Hasil proses *postprocessing* ditunjukkan pada **Gambar 3.19**



Gambar 3.16 Diagram Alir Proses *Clustering*

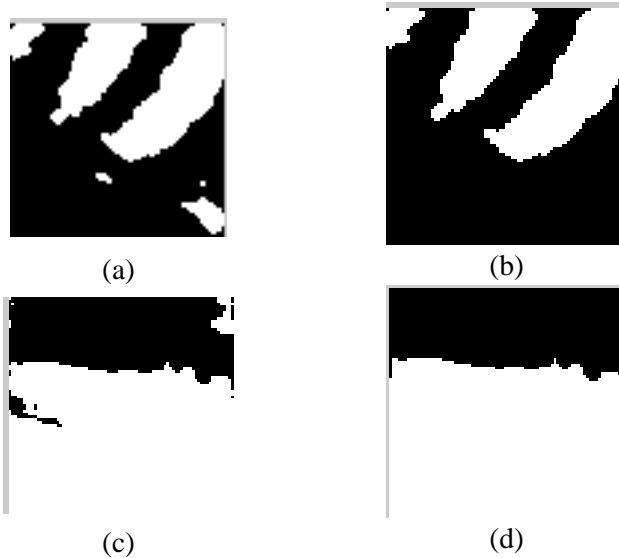


Gambar 3.17 Perbandingan hasil tahap (a) dan (d) *clustering* dan (b) dan (e) tahap klasifikasi, serta (c) dan (f) binerisasi

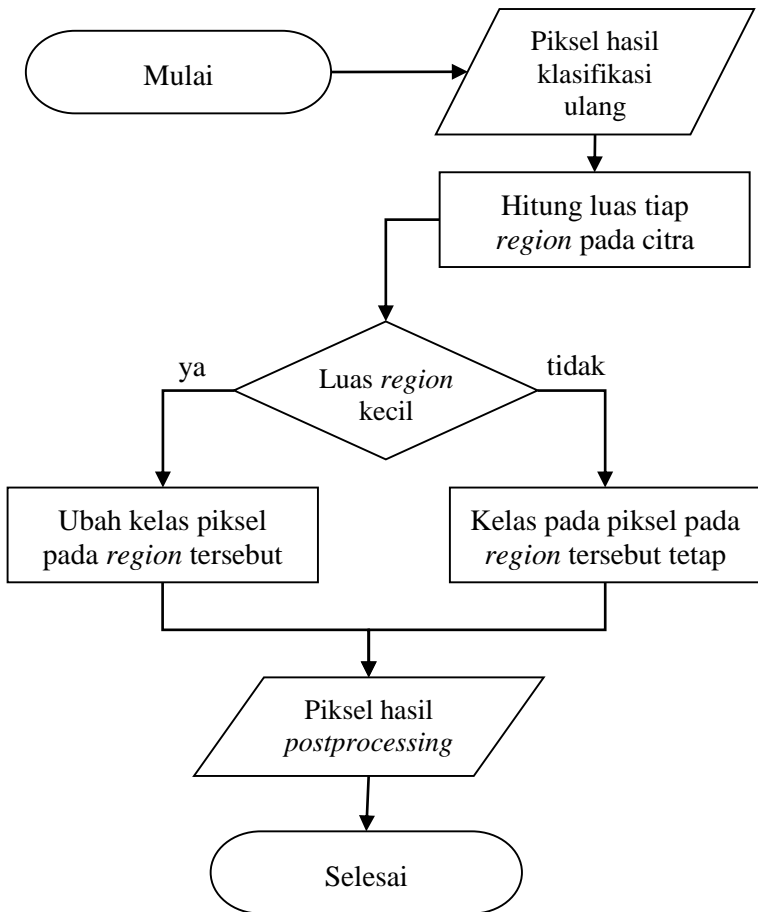


Gambar 3.18 Diagram Alir Proses Klasifikasi

Langkah pertama pada proses ini adalah melakukan pengukuran luas pada *region-region* yang ada pada citra segmentasi. Pada ROI Atas, apabila luas suatu *region* lebih kecil dari 300 piksel², maka *region* tersebut akan dijadikan kelas *trabecular bone*. Sedangkan pada ROI bawah, pertama hasil klasifikasi diinvers dahulu. *Region* yang areanya lebih kecil dari *region* dengan luas maksimal dijadikan kelas *non-trabecular bone*. Setelahnya, citra diinvers lagi dan dilakukan pengukuran luas kembali. *Region* yang areanya lebih kecil dari *region* dengan luas maksimal dijadikan kelas *trabecular bone*.



Gambar 3.19 Perbandingan hasil tahap (a) dan (c) klasifikasi dan (b) dan (d) *postprocessing*



Gambar 3.20 Diagram Alir Proses *Postprocessing*

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III meliputi kode program dalam perangkat lunak. Selain itu, implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program juga dijelaskan.

4.1 Lingkungan Implementasi

Objek citra yang akan diolah pada implementasi tugas akhir ini adalah sebuah citra *Region of Interest* (ROI) dari *scan Dental Panoramic Radiograph* (DPR). Objek citra ROI adalah gambar tipe *bitmap* (bmp).

Dalam implementasi algoritma pengolahan ROI tersebut, digunakan perangkat-perangkat sebagai berikut:

4.1.1 Perangkat Keras

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Perangkat PC yang digunakan adalah tipe desktop *mini tower* bertipe LENOVO H530S dengan sebuah layar monitor.

Spesifikasi dari PC yang digunakan pada tugas akhir ini adalah: memiliki prosesor Intel Core i3 3240 dengan kecepatan 3,4 GHz dan *Random Access Memory* (RAM) untuk proses menjalankan program sebesar 4,00 GB.

4.1.2 Perangkat Lunak

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat lunak menggunakan *software* MATLAB R2014a. Penggunaan MATLAB didukung dengan dua *toolbox* utama yaitu *statistical and machine learning toolbox* dan *image processing toolbox*.

Selain itu, pada tugas akhir ini dalam melakukan pengolahan angka didukung dengan *software* Microsoft Excel. Pembuatan

ground truth dari citra yang ada menggunakan *software* GIMP dan *Microsoft Paint*.

4.2 Implementasi Tahap *Preprocessing*

Tahap *preprocessing* pada tugas akhir ini terdiri dari tiga tahap. Pertama adalah membagi citra menjadi dua, kemudian melakukan normalisasi, melakukan *filtering* (*median filter*, morfologi *structuring element* dan *motion blur*).

Penjelasan dari masing-masing tahapan tersebut adalah sebagai berikut:

4.2.1 Implementasi Pembagian Citra

Tahap ini membagi ROI awal yang berukuran 258x129 menjadi dua ROI berukuran 129x129. Implementasi pembagian citra ditunjukkan pada **Kode Sumber 4.1**. Pada baris 1 dilakukan pembacaan citra dari direktori file. Pada baris 3 dan 4 dilakukan pemotongan ROI sesuai dengan ukuran sumbu Y citra.

ROI dengan piksel sumbu y 1-129 adalah ROI bagian atas, yang terdapat bagian akar gigi. ROI dengan piksel sumbu y 130-258 adalah ROI bagian bawah, yang terdapat bagian *cortical bone*.

1	Citra_asli=imread('path/ke/direktori/citra/dan/nama_file.bmp');
2	[row, col]=size(Citra_asli);
3	ROI_atas=Citra_asli(1:row/2, :);
4	ROI_bawah=Citra_asli(row/2+1:row, :);

Kode Sumber 4.1 Implementasi Pembagian Citra

4.2.2 Implementasi Normalisasi Citra

Normalisasi citra dilakukan dengan tujuan untuk menyamakan rentang nilai piksel satu citra dengan citra lainnya. Formulasi normalisasi ditunjukkan pada Persamaan (2.2).

Implementasi dari pemilihan dan penyimpanan koordinat ROI ditunjukkan pada **Kode Sumber 4.2**. Dihitung nilai intensitas maksimal dan minimal citra pada baris 1 dan 2. Selisih intensitas tiap piksel dihitung pada baris 5 dan rentang piksel pada baris 6.

Penghitungan nilai intensitas normal tiap piksel dilakukan pada baris 7.

1	pmax = max(citra(:));
2	pmin = min(citra(:));
3	for i=1:row
4	for j=1:col
5	a=double(citra(i,j)-pmin);
6	b=double(pmax-pmin);
7	citra(i,j)=a/b*255;

Kode Sumber 4.2 Implementasi Normalisasi Citra

4.2.3 Implementasi *Filtering*

Filtering dilakukan dengan tujuan untuk menghilangkan detail yang mengganggu pada citra dan memperhalus citra. Tahapan dari *filtering* ini adalah *median filter*, morfologi *structuring element* dan *motion blur*.

Implementasi *filtering* ditunjukkan pada **Kode Sumber 4.3**. *Median filter* diimplementasi dengan fungsi `medfilt2` MATLAB pada baris 1. *Structuring element* disusun pada baris 2, dan dilakukan *opening* pada citra pada baris 3. Struktur *motion blur* disusun pada baris 4 dan diterapkan pada baris 5.

1	ROI = medfilt2(ROI);
2	st = strel('line', 5, 90);
3	img90 = imopen(ROI, st);
4	Motion = fspecial('motion', 9, 90);
5	CE_ROI = imfilter(img90, Motion, 'replicate');

Kode Sumber 4.3 Implementasi *Filtering*

4.3 Implementasi Ekstraksi Fitur

Tahap ekstraksi fitur pada tugas akhir ini terdiri dari tiga tahap. Pertama adalah *Gabor filter*, kemudian pengambilan informasi spasial, kemudian pengambilan informasi koordinat sumbu y citra Hasil ekstraksi fitur kemudian dijadikan satu pada suatu matriks.

Penjelasan dari masing-masing tahapan tersebut adalah sebagai berikut:

4.3.1 Implementasi Gabor Filter

Gabor filter digunakan untuk mendapatkan fitur dari ROI yaitu intensitas garis. Implementasi dibagi menjadi dua yaitu untuk perulangan orientasi dan pembentukan *Gabor filter*. Implementasi perulangan orientasi ditunjukkan pada **Kode Sumber 4.4** sementara implementasi pembentukan *Gabor filter* ditunjukkan pada **Kode Sumber 4.5**.

Pada implementasi perulangan orientasi, dipanggil pembuatan filter Gabor pada baris 3, dan diterapkan pada citra pada baris 4. Pada baris 5, hasil filter Gabor digabungkan. Pada baris 6, citra hasil dibagi dengan nilai maksimal citra supaya nilainya menjadi rentang 0-1.

Pada implementasi pembentukan *Gabor filter*, pada baris 1 dihitung nilai sigma. Nilai sigma kemudian menjadi dasar penentuan ukuran dan pembuatan *mask grid* yang disusun pada baris 4 dan 5. Penghitungan x_{θ} dan y_{θ} dilakukan pada baris 6 dan 7, sesuai *mask grid* yang ada. Kemudian diisi nilai filter gabor pada baris 8.

1	<code>img_out = zeros(size(img_in,1), size(img_in,2), N);</code>
2	<code>for n=1:N</code>
3	<code> gb = gabor_fn(bw,gamma,psi,lambda,theta(n));</code>
4	<code> img_out(:,:,n) = imfilter(img_in, gb, 'symmetric');</code>
5	<code> img_out_disp = sum(abs(img_out).^2, 3).^0.5;</code>
6	<code> img_out_disp = img_out_disp./max(img_out_disp(:));</code>

Kode Sumber 4.4 Implementasi Perulangan Orientasi *Gabor Filter*

1	<code>sigma = lambda/pi*sqrt(log(2)/2)*(2^bw+1)/(2^bw-1);</code>
2	<code>sigma_x = sigma;</code>
3	<code>sigma_y = sigma/gamma;</code>
4	<code>sz = fix(5 * max(sigma_y,sigma_x));</code>
5	<code>[x y]=meshgrid(- fix(sz/2):fix(sz/2),fix(sz/2):-1:fix(- sz/2));</code>
6	<code>x_theta=x*cos(theta)+y*sin(theta);</code>
7	<code>y_theta=-x*sin(theta)+y*cos(theta);</code>
8	<code>gb=exp(- 0.5*(x_theta.^2/sigma_x^2+y_theta.^2/sigma_y^2)).*cos(2*pi/lambda*x_theta+psi);</code>

Kode Sumber 4.5 Implementasi Pembentukan *Gabor Filter*

4.3.2 Implementasi Pengambilan Fitur Spasial

Fungsi ini digunakan untuk mendapatkan fitur dari ROI yaitu rata-rata nilai piksel dari tetangga piksel dalam *window* berukuran nxn. Pada citra dilakukan *padding* terlebih dahulu pada tepi citra, kemudian dilakukan perhitungan fitur spasial.

Pada baris dilakukan *padding* dengan cara mereplikasi matriks. Setelah itu, nilai tetangga piksel dirata-rata pada baris 7. Nilai jumlah piksel dinormalkan pada baris 11. Implementasi pengambilan fitur spasial ditunjukkan pada **Kode Sumber 4.6**

1	<code>neighborinfo = ones(129, 129);</code>
2	<code>pixdata2=padarray(pixdata,[fix(n/2) fix(n/2)], 'replicate');</code>
3	<code>for i=1+fix(n/2):129+fix(n/2)</code>
4	<code>for ii=1+fix(n/2):129+fix(n/2)</code>
5	<code>for j=1:n</code>
6	<code>for k=1:n</code>
7	<code> window(j,k) = pixdata2(i+j-fix(n/2)- 1, ii+k-fix(n/2)-1);</code>
8	<code> neighborinfo(i,ii)=mean(window(:));</code>
9	<code>for i=1:129</code>

Kode Sumber 4.6 Implementasi Pengambilan Fitur Spasial

10	for j=1:129
11	spatialinfo=neighborinfo/max(neighborinfo(:));

Kode Sumber 4.6 Implementasi Pengambilan Fitur Spasial (Lanjutan)

4.3.3 Implementasi Pengambilan Koordinat Sumbu Y

Fungsi ini digunakan untuk mendapatkan fitur dari ROI yaitu koordinat sumbu Y piksel. Fungsi ini dijalankan khusus untuk ROI bagian bawah.

Pada baris 4 ditunjukkan cara pengambilan nilai koordinat sumbu Y untuk setiap piksel pada ROI. Implementasi pengambilan fitur spasial ditunjukkan pada **Kode Sumber 4.7**

1	koord_y=zeros(129,129);
2	for i=1:129
3	for j=1:129
4	koord_y(i,j)=(i-1)/128;
5	koord_y=zeros(129,129);

Kode Sumber 4.7 Implementasi Pengambilan Koordinat Sumbu Y

4.4 Implementasi Proses *Clustering*

Tahap *clustering* pada tugas akhir ini terdiri dari dua tahap. Pertama adalah *clustering* dan binerisasi. Kedua adalah pengukuran jarak tiap piksel terhadap *centroid* dan *thresholding* piksel yang dianggap jaraknya terlalu jauh dari *centroid*.

Penjelasan dari masing-masing tahapan tersebut adalah sebagai berikut:

4.4.1 Implementasi Proses *K-means Clustering*

K-means clustering dilakukan dengan tujuan untuk mengelompokkan piksel-piksel pada citra menjadi kelas area *trabecular bone* dan area *non-trabecular bone*. Setelah dilakukan *clustering* sekaligus dilakukan binerisasi pada kelompok piksel yang dihasilkan.

Implementasi dibagi menjadi tiga yaitu untuk *K-means clustering* dan pemberian warna untuk ROI atas, *K-means clustering* dan pemberian warna untuk ROI bawah, serta implementasi inisialisasi *centroid*. Proses pemberian warna dipisah karena perbedaan jumlah *cluster*. Implementasi *K-means clustering* dan pemberian warna untuk ROI atas ditunjukkan pada **Kode Sumber 4.8**, implementasi *K-means clustering* dan pemberian warna untuk ROI atas ditunjukkan pada **Kode Sumber 4.9**, dan implementasi penentuan *centroid* ditunjukkan pada **Kode Sumber 4.10**. Pada implementasi inisialisasi *centroid*, baris 1 melakukan sampling *uniform*, dan baris 2-3 merubah *centroid* yang didapat menjadi data *double*. Sedangkan pada kode sumber lainnya, baris 1 adalah pengimplementasian *K-means clustering* dan baris lainnya adalah implementasi pemberian warna.

	[cluster_atas, C_aki] =
1	kmeans(data_fitur_atas, 2);
2	for i=1:length(cluster_atas)
3	if cluster_atas(i) == 1
4	if C_aki(1,1)>C_aki(2,1)
5	clusres(i)=1;
6	if C_aki(2,1)>C_aki(1,1)
7	clusres(i)=0;
8	else
9	if C_aki(1,1)>C_aki(2,1)
10	clusres(i)=0;
11	if C_aki(2,1)>C_aki(1,1)
12	clusres(i)=1;

Kode Sumber 4.8 Implementasi *K-means Clustering* dan Pemberian Warna ROI Atas

	[cluster_kiri_atas, C_bki] =
1	kmeans(data_fitur_kiri_atas, 3);
2	[~,max_ki]=max(C_bki(:));
3	[~,min_ki]=min(C_bki(:));
4	for i=1:length(cluster_kiri_atas)
5	if cluster_kiri_atas(i) == 1
6	if max(C_bki(:,1))==C_bki(1,1)

Kode Sumber 4.9 Implementasi *K-means Clustering* dan Pemberian Warna ROI Bawah

7	clusres(i)=0.5;
8	if max(C_bki(:,1))~=C_bki(1,1) && min(C_bki(:,1))~=C_bki(1,1)
9	clusres(i)=1;
10	if min(C_bki(:,1))==C_bki(1,1)
11	clusres(i)=0;
12	if cluster_kiri_atas(i) == 2
13	if max(C_bki(:,1))==C_bki(2,1)
14	clusres(i)=0.5;
15	if max(C_bki(:,1))~=C_bki(2,1) && min(C_bki(:,1))~=C_bki(2,1)
16	clusres(i)=1;
17	if min(C_bki(:,1))==C_bki(2,1)
18	clusres(i)=0;
19	if cluster_kiri_atas(i) == 3
20	if max(C_bki(:,1))==C_bki(3,1)
21	clusres(i)=0.5;
22	if max(C_bki(:,1))~=C_bki(3,1) && min(C_bki(:,1))~=C_bki(3,1)
23	clusres(i)=1;
24	if min(C_bki(:,1))==C_bki(3,1)
25	clusres(i)=0;

Kode Sumber 4.9 Implementasi *K-means Clustering* dan Pemberian Warna ROI Bawah (Lanjutan)

1	C = data_fitur(randsample(S,row,k),:);
2	if ~isfloat(C)
3	C = double(C);

Kode Sumber 4.10 Implementasi Inisialisasi *Centroid*

4.4.2 Implementasi Penghitungan Jarak Pixel ke *Centroid* dan *Thresholding*

Proses ini dilakukan untuk mendapatkan jarak dari suatu pixel ke *centroid* dari *cluster* tempat pixel tersebut menjadi anggota. Jarak antara pixel dengan *centroid* dihitung dengan menggunakan *euclidian distance*. Kemudian ditentukan secara manual suatu nilai *threshold* untuk membatasi pixel yang

dianggap terlalu jauh dari centroid. Pixel yang jaraknya melebihi *threshold* kemudian disimpan pada suatu matriks sebagai pixel yang akan diklasifikasikan ulang pada proses selanjutnya.

Implementasi pengukuran jarak dan *thresholding* ditunjukkan pada **Kode Sumber 4.11**. Pada baris 3-6 dan 9-12 dihitung jarak *euclidean* tiap pixel. Pada baris 7-8 dan 13-14 dihitung nilai *threshold*. Pada baris 15-20 diseleksi pixel yang jaraknya ada di atas *threshold*.

1	[row1,~]=size(satu);
2	[row2,~]=size(dua);
3	for i=1:row1
4	jarakk = ((abs(satu(i,1)-C_aki(ichi,1)).^2) + (abs(satu(i,2)-C_aki(ichi,2)).^2) + (abs(satu(i,3)-C_aki(ichi,3)).^2) + (abs(satu(i,4)-C_aki(ichi,4)).^2)).^0.5;
5	dist_satu(i,1) = jarakk;
6	dist_satu(i,2) = satu(i,5);
7	max_satu=max(dist_satu);
8	thresh_satu=threshold*max_satu(1);
9	for i=1:row2
10	jarakk = ((abs(dua(i,1)-C_aki(ni,1)).^2) + (abs(dua(i,2)-C_aki(ni,2)).^2) + (abs(dua(i,3)- C_aki(ni,3)).^2) + (abs(dua(i,4)- C_aki(ni,4)).^2)).^0.5;
11	dist_dua(i,1) = jarakk;
12	dist_dua(i,2) = dua(i,5);
13	max_dua=max(dist_dua);
14	thresh_dua=threshold*max_dua(1);
15	for i=1:row1
16	if dist_satu(i,1) > thresh_satu
17	ceklagi=vertcat(ceklagi, dist_satu(i,2));
18	for i=1:row2
19	if dist_dua(i,1) > thresh_dua
20	ceklagi=vertcat(ceklagi, dist_dua(i,2));

Kode Sumber 4.11 Implementasi Penghitungan Jarak Centroid dan Thresholding

4.5 Implementasi Proses Klasifikasi

Tahap klasifikasi pada tugas akhir ini terdiri dari dua tahap. Pertama dilakukan proses *training* pada *decision tree* menggunakan metode *Root-Guided Decision Tree* (RGDT), dan kedua dilakukan klasifikasi ulang piksel hasil *clustering* menggunakan *decision tree* yang dihasilkan.

Penjelasan dari masing-masing tahapan tersebut adalah sebagai berikut:

4.5.1 Implementasi *Training Decision Tree*

Proses *training decision tree* dilakukan untuk membentuk suatu *decision tree* sebagai *rule* untuk melakukan klasifikasi. Algoritma yang digunakan untuk menyusun *decision tree* adalah RGDT. *Decision tree* yang dihasilkan disimpan dalam bentuk file .mat. Implementasi RGDT dibagi menjadi dua yaitu perulangan penyusunan *tree* tiap atribut dan penyusunan *tree* secara khusus.

Implementasi perulangan penyusunan *tree* untuk setiap atribut ditunjukkan pada **Kode Sumber 4.12** dan implementasi penyusunan *tree* secara khusus ditunjukkan pada **Kode Sumber 4.13**. Pada penyusunan *tree* untuk tiap atribut, pada baris 1-3 disusun *decision tree*. Pada baris 4-6 diuji performa *tree* tiap atribut, dan akurasi dihitung pada baris 8-17. Baris 18-26 digunakan untuk menyimpan *tree* yang terbaik. Pada penyusunan *tree* secara khusus, pada baris 1-3 dihitung nilai entropi untuk *node*. Pada baris 4-19 dicari *split location* terbaik untuk *node* tersebut. Pada baris 20-26 diberi kondisi apabila *node* adalah *root*, maka *split location* diatur terletak pada atribut sesuai aturan RGDT. Pada baris 30-32 data yang ada dipecah sesuai dengan *split location*. Pada baris 33-38 mengatur isi *node* jika *node* merupakan *leaf*, sedangkan baris 39-41 mengatur isi *node* jika *node* bukan *leaf* dan merekursi fungsi untuk menyusun *child* dari *node* itu.

1	<pre>tree = C4_5TrainFun(train_patterns, train_targets, inc_node, depth, prune, 1, isroot);</pre>
---	---

Kode Sumber 4.12 Implementasi Penyusunan *Tree* untuk Tiap Atribut

2	tree2 = C4_5TrainFun(train_patterns, train_targets, inc_node, depth, prune, 2, isroot);
3	tree3 = C4_5TrainFun(train_patterns, train_targets, inc_node, , depth, prune, 3, isroot);
4	test_targets = C4_5TestFun(test_patterns, tree);
5	test_targets2 = C4_5TestFun(test_patterns, tree2);
6	test_targets3 = C4_5TestFun(test_patterns, tree3);
7	[row, col] = size(ground_truth);
8	for i = 1 : col
9	if test_targets(1,i) == ground_truth(1,i)
10	benar=benar+1;
11	if test_targets2(1,i) == ground_truth(1,i)
12	benar2=benar2+1;
13	if test_targets3(1,i) == ground_truth(1,i)
14	benar3=benar3+1;
15	akurasi2(1,1) = benar/col*100;
16	akurasi2(2,1) = benar2/col*100;
17	akurasi2(3,1) = benar3/col*100;
18	[maksimale, tree_ke]=max(akurasi2(:));
19	if maksimale>maksimale_temp
20	if tree_ke==1
21	treeres=tree;
22	if tree_ke==2
23	treeres=tree2;
24	if tree_ke==3
25	treeres=tree3;
26	maksimale_temp=maksimale;

Kode Sumber 4.12 Implementasi Penyusunan *Tree* untuk Tiap Atribut (Lanjutan)

1	for i = 1:length(Uc),
	Pnode(i) = length(find(targets == Uc(i)))
2	/ L;
3	Inode = -sum(Pnode.*log(Pnode)/log(2));
4	for i = 1:Ni,
5	P = zeros(length(Uc), 2);
6	[sorted_data, indices] = sort(data);
7	sorted_targets = targets(indices);
8	I = zeros(1, L-1);
9	for j = 1:L-1,
10	for k = 1:length(Uc),
	P(k,1) =
11	length(find(sorted_targets(1:j) ==
	Uc(k)));
	P(k,2) =
12	length(find(sorted_targets(j+1:end) ==
	Uc(k)));
13	Pk = sum(P);
14	P = P./(eps+repmat(Pk,size(P,1),1));
15	Pk = Pk/sum(Pk);
16	Info = sum(-P.*log(eps+P)/log(2));
	I(j) + (Inode - sum(Pk.*info)) / -
17	sum(Pk.*log(eps+Pk)/log(2));
18	[delta Ib(i), s] = max(I);
19	split_loc(i) = sorted_data(s);
20	if isroot == 1
21	m = delta Ib(attribke);
22	dim = attribke;
23	isroot = 0;
24	else
25	[m, dim] = max(delta Ib);
26	dims = 1:Ni;
27	tree.dim = dim;
28	Nf = unique(patterns(dim,:));
29	Nbins = length(Nf);
30	tree.split_loc = split_loc(dim);
31	indices1 = find(patterns(dim,:) <=
	split_loc(dim));

Kode Sumber 4.13 Implementasi Penyusunan *Tree* secara Khusus

32	<code>indices2 = find(patterns(dim,:)) ></code> <code>split_loc(dim));</code>
32	<code>if (isempty(indices1)) </code> <code>(isempty(indices2))</code>
33	<code>tempvec1=zeros(1,length(Uc));</code>
34	<code>for ii=1:length(Uc)</code>
35	<code>tempvec1(ii)=length(find(targets==Uc(ii)))</code> <code>);</code>
36	<code>[m, largest] = max(tempvec1);</code>
37	<code>tree.child = Uc(largest);</code>
38	<code>return</code>
39	<code>else</code>
40	<code>tree.child(1) =</code> <code>C4_5TrainFun(patterns(dims2, indices1),</code> <code>targets(indices1), inc_node, depth,</code> <code>prune, attribke, isroot);</code>
41	<code>tree.child(2) =</code> <code>C4_5TrainFun(patterns(dims2, indices2),</code> <code>targets(indices2), inc_node, depth,</code> <code>prune, attribke, isroot);</code>

Kode Sumber 4.13 Implementasi Penyusunan *Tree* secara Khusus (Lanjutan)

4.5.2 Implementasi Klasifikasi Ulang Piksel

Proses klasifikasi ulang piksel dilakukan untuk mengklasifikasi kembali piksel-piksel yang masuk kategori klasifikasi ulang pada hasil *K-means clustering*. *Decision tree* yang digunakan untuk klasifikasi adalah *decision tree* yang dihasilkan dari proses RGDT. *Decision tree* tersebut di-load dari file .mat yang dihasilkan dari RGDT.

Implementasi klasifikasi ulang piksel ditunjukkan pada **Kode Sumber 4.14**. Pada baris 1 di-load *decision tree* hasil *training*. Baris 2-5 mengecek apabila *node* adalah *leaf* maka menghentikan fungsi (*return*). Baris 5-10 mengecek apabila *node* bukan *leaf* maka dilakukan rekursi ke *child* dari *node* itu.

1	<code>load tree_RGDT</code>
2	<code>if (length(tree.child) == 1)</code>

Kode Sumber 4.14 Implementasi Klasifikasi Ulang Piksel

3	<code>targets(indices) = tree.child;</code>
4	<code>return</code>
5	<code>dim = tree.dim;</code>
6	<code>dims= 1:size(patterns,1);</code>
7	<code>in = indices(find(patterns(dim, indices) <=</code> <code>tree.split_loc));</code>
8	<code>targets(in) = 4_5TestFun(patterns(dims, in),</code> <code>tree.child(1), discrete_dim(dims));</code>
9	<code>in = indices(find(patterns(dim, indices) ></code> <code>tree.split_loc));</code>
10	<code>targets(in)= C4_5TestFun(patterns(dims,in),</code> <code>tree.child(2), discrete_dim(dims));</code>

**Kode Sumber 4.14 Implementasi Klasifikasi Ulang Piksel
(Lanjutan)**

4.5.3 Implementasi *Postprocessing*

Proses *postprocessing* dilakukan untuk memperbaiki citra yang dihasilkan oleh proses klasifikasi. Pada tahap ini dilakukan pengukuran luas *region* pada citra segmentasi dan perubahan kelas piksel pada *region* tersebut.

Implementasi *postprocessing* ditunjukkan pada **Kode Sumber 4.15**. Pada baris 1 dilakukan pengukuran luas *region*. Pada baris 2-6 dilakukan seleksi piksel pada *region* yang memiliki luas kecil. Pada baris 7-10 dilakukan pengubahan kelas piksel pada *region* yang memiliki luas kecil.

	<code>stats=regionprops(im2bw(imres_dt),</code>
1	<code>'Area', 'PixelIdxList');</code>
2	<code>for i=1:numel(stats)</code>
3	<code>areas=vertcat(areas, stats(i).Area);</code>
4	<code>for i=1:numel(stats)</code>
5	<code>if areas(i+1)<limit</code>
6	<code>pp=vertcat(pp,stats(i).PixelIdxList);</code>
7	<code>if row pp>1</code>
8	<code>pp=pp(2:end);</code>
9	<code>for i=1:row pp-1</code>
10	<code>imres_dt2(pp(i))=0;</code>

Kode Sumber 4.15 Implementasi *Postprocessing*

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan.

5.1 Lingkungan Uji Coba

Lingkungan uji coba pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi PC dari sisi perangkat keras adalah memiliki prosesor Intel Core i3 3240 dengan kecepatan 3,4 GHz dan memori untuk proses sebesar 4,00 GB. PC yang digunakan memiliki sistem operasi Windows 8.

Pada sisi perangkat lunak, uji coba pada tugas akhir ini dilakukan dengan menggunakan *software* MATLAB R2014a. Penggunaan MATLAB didukung dengan dua *toolbox* utama yaitu *statistical and machine learning toolbox* dan *image processing toolbox*. Dokumentasi hasil uji coba dilakukan dengan menggunakan *Microsoft Paint*.

5.2 Data Uji Coba

Data uji coba yang digunakan sebagai masukan adalah citra *dental panoramic radiograph* (DPR). Contoh DPR ditunjukkan pada **Gambar 5.1**. Dari DPR dilakukan proses *cropping* untuk mendapatkan 30 buah *Region of Interest* (ROI) yang memuat area *trabecular bone*. Ukuran ROI yang digunakan adalah 256×128 piksel. Contoh hasil dari proses *cropping* ditunjukkan pada

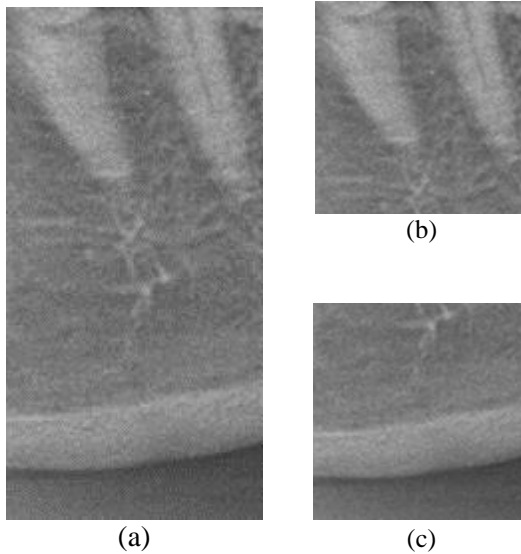
Untuk menguji kebenaran dari hasil segmentasi, digunakan data *ground truth* berupa citra hitam putih. Data *ground truth* diperoleh dengan cara pembuatan manual dari masing-masing ROI menggunakan *tool* GIMP oleh pengamat yang sama. Contoh data *ground truth* ditunjukkan pada **Gambar 5.3**

Untuk penyusunan *Root-Guided Decision Tree* (RGDT) dalam uji coba, digunakan 20.000 piksel yang dipilih acak dengan *sampling with replacement* dari keseluruhan piksel pada 30 buah

ROI. *Decision tree* yang dihasilkan lalu dilakukan *testing* pada keseluruhan ROI.



Gambar 5.1 Citra *Dental Panoramic Radiograph*



Gambar 5.2 Citra Hasil *Cropping*
(a) ROI asli, (b) ROI atas, dan (c) ROI bawah



Gambar 5.3 Citra *Ground Truth*

5.3 Skenario Uji Coba

Uji coba dilakukan untuk mengetahui nilai-nilai parameter yang tepat untuk digunakan pada masing-masing proses. Nilai parameter yang tepat penting untuk diketahui karena penggunaan parameter yang tepat akan memberikan hasil yang terbaik pada keluaran tiap proses.

Skenario pengujian terdiri dari lima macam yaitu:

1. Uji coba penentuan parameter *Gabor filter* (orientasi, *wavelength*, *bandwith*, dan *gamma*).
2. Uji coba penentuan ukuran *window* fitur spasial
3. Uji coba penentuan nilai *threshold*
4. Uji coba penentuan level *pruning* pada *Root-Guided Decision Tree* (RGDT).
5. Uji coba menggunakan metode pembandingan

5.4 Uji Coba Penentuan Parameter pada *Gabor Filter*

Uji coba penentuan parameter pada *Gabor filter* diperlukan untuk mendapatkan hasil ekstraksi fitur yang baik dan sesuai kebutuhan. Hasil ekstraksi fitur yang baik akan memberikan hasil *clustering* yang baik pada proses selanjutnya.

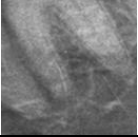

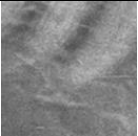

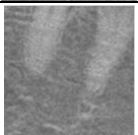

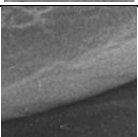

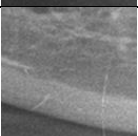

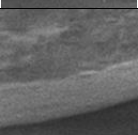

Beberapa parameter yang terdapat pada *Gabor filter* adalah: sudut, *wavelength*, *phase offset*, *bandwith*, dan *gamma*. Masing-masing parameter mempengaruhi hasil keluaran dari *Gabor filter* pada aspek yang berbeda-beda.

5.4.1 Uji Coba Penentuan Parameter Sudut pada *Gabor Filter*

Uji coba penentuan parameter sudut pada *Gabor filter* diperlukan agar hasil ekstraksi fitur yang ditemukan sesuai dengan kebutuhan. Tujuannya untuk menemukan nilai sudut yang sesuai dengan orientasi objek *non-trabecular bone* pada citra yaitu akar gigi untuk ROI atas dan *cortical bone* untuk ROI bawah.

Penentuan parameter dipisah untuk ROI atas dan ROI bawah. Untuk ROI atas dicari nilai sudut yang menyerupai orientasi objek akar gigi, dan untuk ROI bawah dicari nilai sudut yang menyerupai orientasi objek *cortical bone*. Pengujian dilakukan dengan mendapatkan nilai orientasi pada beberapa sampel *Ground truth* dari ROI atas dan ROI bawah dengan menggunakan fungsi *regionprops* pada MATLAB. Hasil uji coba ditunjukkan pada **Tabel 5.1**. Dari tabel tersebut dapat disimpulkan bahwa orientasi objek *non-trabecular bone* pada ROI atas adalah $50,00^{\circ}$, $72,09^{\circ}$, $113,25^{\circ}$ dan orientasi objek *non-trabecular bone* pada ROI bawah adalah $7,83^{\circ}$, $21,12^{\circ}$, $162,80^{\circ}$. Jika dibulatkan untuk setiap interval 30° maka didapat sudut pada ROI atas adalah 60° , dan 120° sedangkan pada ROI bawah adalah 0° , 30° , 150°

Tabel 5.1 Hasil Uji Coba Parameter Sudut

ROI	Citra Asli ROI	Ground Truth ROI	Orientasi Objek non-trabecular bone (derajat)
ROI Atas 3			113,25
ROI Atas 4			50,00
ROI Atas 22			72,09
ROI Bawah 5			21,13
ROI Bawah 8			162,80
ROI Bawah 9			7,83

5.4.2 Uji Coba Penentuan Parameter *Wavelength* pada *Gabor Filter*

Uji coba penentuan parameter *wavelength* pada *Gabor filter* diperlukan agar hasil ekstraksi fitur yang dilakukan sesuai dengan kebutuhan. Hal ini dilakukan karena nilai parameter *wavelength* dari *Gabor filter* menentukan ukuran filter.

Nilai *wavelength* yang digunakan dalam uji coba adalah 1, 3, 5, 7, 9. Pengukuran keberhasilan parameter *wavelength* didapat dari nilai akurasi apabila dilakukan *clustering* dengan data fitur menggunakan nilai *Gabor filter* dengan parameter tetap sudut 60^0 untuk ROI atas dan 0^0 untuk ROI bawah, *phase offset* 0, *bandwith* 2 dan *gamma* 3. Hasil uji coba ditunjukkan pada **Tabel 5.2**, **Tabel 5.3**, **Gambar 5.4**, dan **Gambar 5.5**. Dari tabel tersebut terlihat bahwa dua nilai *wavelength* terbaik adalah 1 dan 3 untuk ROI atas dan ROI bawah.

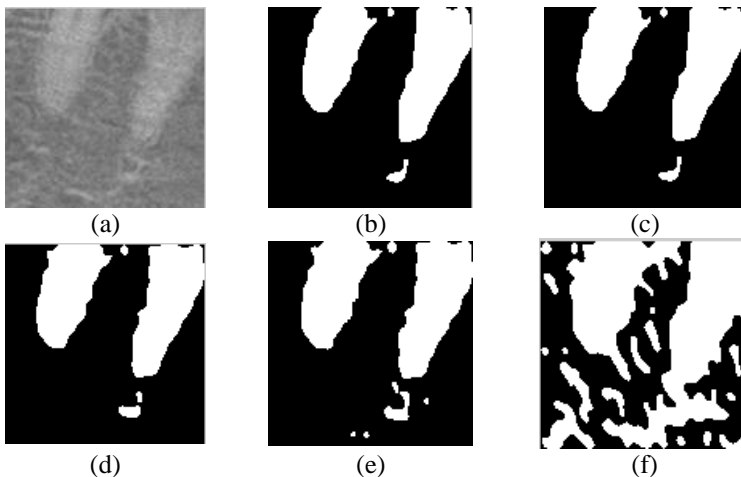
Wavelength rendah menghasilkan gambar yang halus dan *wavelength* tinggi menghasilkan gambar yang peka terhadap detail termasuk tepi objek dalam ROI. Agar data fitur yang dihasilkan representatif terhadap objek dalam citra, dilakukan beberapa kali filter Gabor dengan *wavelength* berbeda. Dipilih *wavelength* dari 1 sampai 3, karena *wavelength* 1 dan *wavelength* 3 adalah dua nilai dengan hasil akurasi *cluster* tertinggi, dan secara visual tidak banyak menimbulkan kesalahan klasifikasi dan tidak memperkuat *noise* dibandingkan nilai *wavelength* di atasnya.

Tabel 5.2 Hasil Uji Coba Parameter *Wavelength* pada ROI Atas

<i>Wavelength</i>	Akurasi <i>Cluster</i> (%)
1	97,00
3	96,96
5	96,77
7	96,05
9	75,78

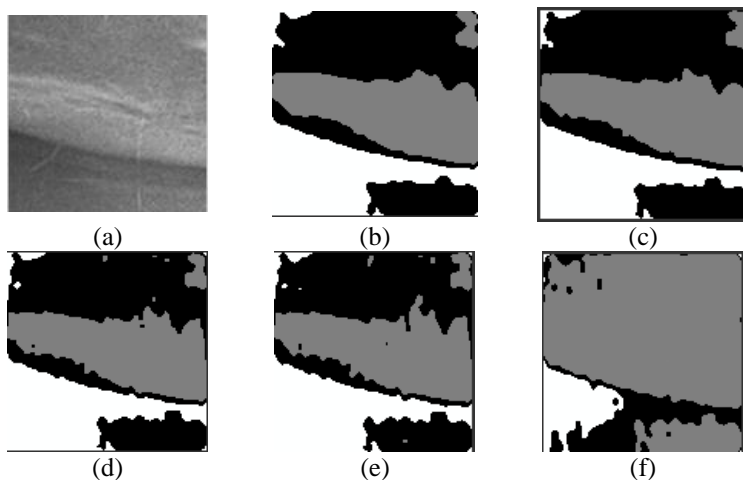
Tabel 5.3 Hasil Uji Coba Parameter *Wavelength* pada ROI Bawah

<i>Wavelength</i>	Akurasi <i>Cluster</i> (%)
1	74,30
3	74,27
5	74,26
7	73,64
9	38,79

Gambar 5.4 Hasil Uji Coba Parameter *Wavelength* pada ROI Atas
(a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan parameter *wavelength* (b) 1, (c) 3, (d) 5, (e) 7, (f) 9

5.4.3 Uji Coba Penentuan Parameter *Bandwith* pada *Gabor Filter*

Uji coba penentuan parameter *bandwith* pada *Gabor filter* diperlukan agar hasil ekstraksi fitur yang dilakukan sesuai dengan kebutuhan. Hal ini dilakukan karena nilai parameter *bandwith* dari *Gabor filter* menentukan ukuran filter.



Gambar 5.5 Hasil Uji Coba Parameter *Wavelength* pada ROI Bawah

(a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan parameter *wavelength* (b) 1, (c) 3, (d) 5, (e) 7, (f) 9

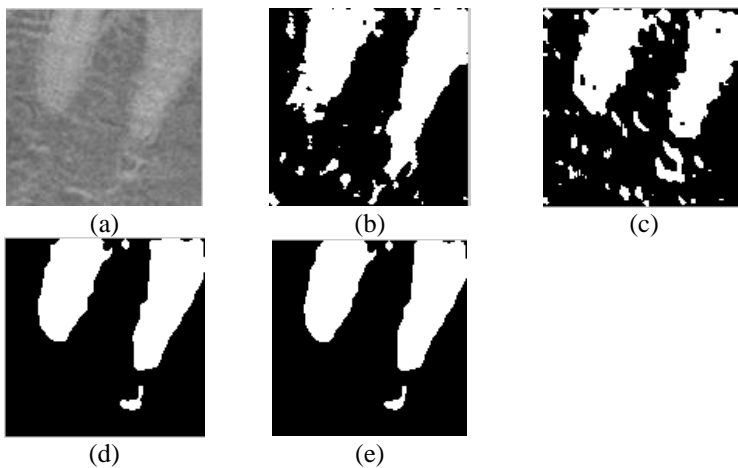
Nilai *bandwith* yang digunakan dalam uji coba adalah 0.5, 1, 1.5, dan 2. Pengukuran keberhasilan parameter *bandwith* didapat dari nilai akurasi apabila dilakukan *clustering* dengan data fitur menggunakan nilai *Gabor filter* dengan parameter tetap sudut 60^0 untuk ROI atas dan 0^0 untuk ROI bawah, *wavelength* 3, *phase offset* 0, dan *gamma* 3. Hasil uji coba ditunjukkan pada **Tabel 5.4**, **Tabel 5.5**, **Gambar 5.6**, dan **Gambar 5.7**. Dari tabel tersebut terlihat bahwa nilai *bandwith* terbaik adalah 2 untuk ROI atas dan ROI bawah.

Tabel 5.4 Hasil Uji Coba Parameter *Bandwith* pada ROI Atas

<i>Bandwith</i>	Akurasi Cluster (%)
0,5	89,47
1	90,28
1,5	96,84
2	96,96

Tabel 5.5 Hasil Uji Coba Parameter *Bandwith* pada ROI Bawah

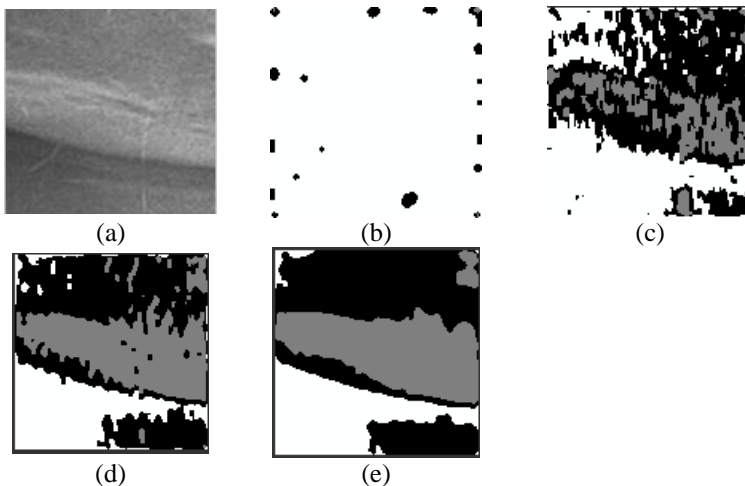
<i>Bandwith</i>	Akurasi Cluster (%)
0,5	31,09
1	69,45
1,5	71,11
2	74,27



Gambar 5.6 Hasil Uji Coba Parameter *Bandwith* pada ROI Atas
 (a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan parameter *bandwith* (b) 0,5, (c) 1, (d) 1,5, (e) 2

5.4.4 Uji Coba Penentuan Parameter *Gamma* pada *Gabor Filter*

Uji coba penentuan parameter *gamma* pada *Gabor filter* diperlukan agar hasil ekstraksi fitur yang dilakukan sesuai dengan kebutuhan. Hal ini dilakukan karena nilai parameter *gamma* dari *Gabor filter* menentukan bentuk dan ukuran filter.



Gambar 5.7 Hasil Uji Coba Parameter *Bandwith* pada ROI Bawah
 (a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan parameter *bandwith* (b) 0,5, (c) 1, (d) 1,5, (e) 2

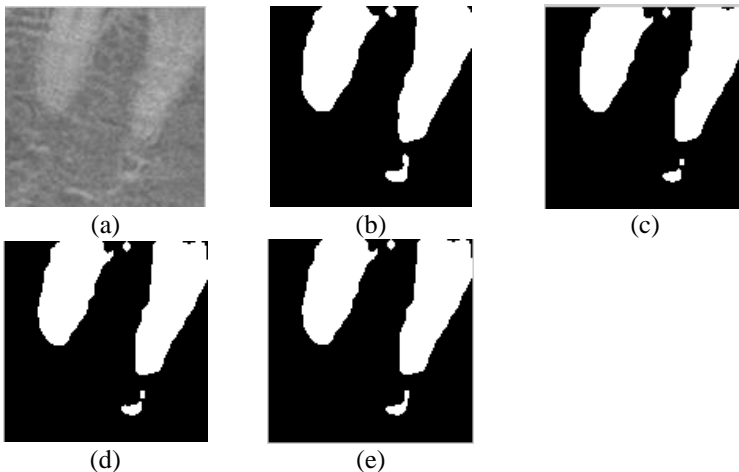
Nilai *bandwith* yang digunakan dalam uji coba adalah 0,5, 1, 1,5, dan 2. Pengukuran keberhasilan parameter *bandwith* didapat dari nilai akurasi apabila dilakukan *clustering* dengan data fitur menggunakan nilai *Gabor filter* dengan parameter tetap sudut 60^0 untuk ROI atas dan 0^0 untuk ROI bawah, *wavelength* 3, *phase offset* 0, dan *bandwith* 2. Hasil uji coba ditunjukkan pada **Tabel 5.6**, **Tabel 5.7**, **Gambar 5.8** dan **Gambar 5.9**. Dari tabel tersebut terlihat bahwa nilai *gamma* terbaik adalah 2 untuk ROI atas dan 3 untuk ROI bawah.

Tabel 5.6 Hasil Uji Coba Parameter *Gamma* pada ROI Atas

<i>Gamma</i>	Akurasi Cluster (%)
0,5	96,83
1	96,93
2	96,99
3	96,96

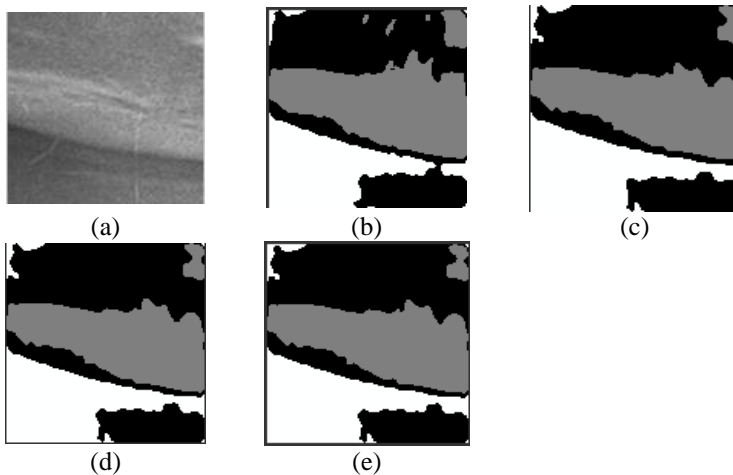
Tabel 5.7 Hasil Uji Coba Parameter *Gamma* pada ROI Bawah

<i>Gamma</i>	Akurasi <i>Cluster</i> (%)
0,5	70,72
1	74,20
2	74,13
3	74,27

**Gambar 5.8 Hasil Uji Coba Parameter *Gamma* pada ROI Atas**
(a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan parameter *gamma* (b) 0,5, (c) 1, (d) 2, (e) 3

5.5 Uji Coba Fitur Spasial

Uji coba ini berfungsi untuk mengetahui efek penambahan fitur spasial terhadap hasil *clustering* pada ROI. Pengujian ini dilakukan dengan membandingkan nilai akurasi apabila fitur spasial digunakan pada *clustering* dan jika fitur spasial tidak digunakan pada *clustering*. Fitur spasial akan ditambahkan pada data fitur dasar berupa tiga citra hasil filter Gabor dengan masing-masing wavelength 1, 2, 3 dan parameter lain sesuai dengan yang telah diujicobakan sebelumnya.



Gambar 5.9 Hasil Uji Coba Parameter *Gamma* pada ROI Bawah
(a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan parameter *gamma* (b) 0,5, (c) 1, (d) 2, (e) 3

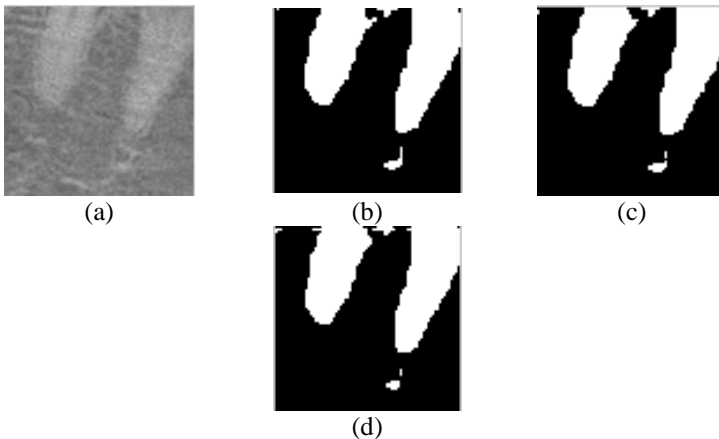
Uji coba dilakukan untuk mengetahui ukuran *window* yang paling optimal untuk mendapatkan fitur spasial rata-rata nilai piksel tetangga. Hasil uji coba ditunjukkan pada **Tabel 5.8**, **Tabel 5.9**, **Gambar 5.10**, dan **Gambar 5.11**. Dari tabel tersebut terlihat bahwa penggunaan fitur spasial jumlah nilai piksel tetangga dapat memperbaiki akurasi *clustering* ROI atas dan penggunaan fitur spasial jumlah nilai piksel tetangga dan koordinat sumbu y dapat memperbaiki akurasi *clustering* ROI bawah.

Tabel 5.8 Hasil Uji Coba Fitur Spasial pada ROI Bawah

Ukuran Window (piksel)	Akurasi Cluster (%)
3x3	96,96
5x5	97,08
7x7	97,12

Tabel 5.9 Hasil Uji Coba Fitur Spasial pada ROI Atas

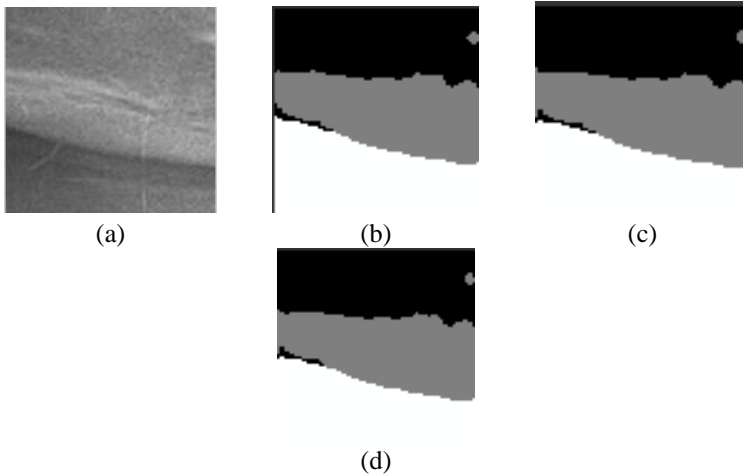
Ukuran Window (piksel)	Akurasi Cluster (%)
3x3	88,47
5x5	88,69
7x7	88,95



Gambar 5.10 Hasil Uji Coba Fitur Spasial pada ROI Atas
 (a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan fitur spasial dari *window* berukuran (a) 3x3, (b) 5x5, dan (c) 7x7

5.6 Uji Coba Penentuan Nilai *Threshold* pada Hasil *K-means Clustering*

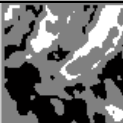


Uji coba ini dilakukan untuk mengetahui nilai *threshold* yang optimal untuk menentukan batas bahwa suatu piksel dianggap jauh atau dekat dari *centroid cluster* tempat piksel tersebut berada. Hal ini penting ditentukan karena piksel yang berada di atas nilai *threshold* dianggap menjadi piksel yang salah klasifikasi pada *clustering* dan akan menjadi data masukan pada proses klasifikasi ulang. Piksel di atas *threshold* diberi warna abu-abu untuk ROI atas dan abu-abu tua untuk ROI bawah.






Gambar 5.11 Hasil Uji Coba Fitur Spasial pada ROI Atas
(a) Gambar asli, Hasil *clustering* dengan filter Gabor dengan fitur spasial dari *window* berukuran (a) 3x3, (b) 5x5, dan (c) 7x7

Pengujian dilakukan dengan menghitung nilai *F1 Score* pada piksel yang berada di atas *threshold*. *F1 Score* digunakan untuk mengetahui seberapa baik nilai *threshold* dapat menemukan piksel yang salah klasifikasi berdasarkan *precision* dan *recall*. Untuk pengujian ini, *true positive* adalah piksel yang berada di atas *threshold* dan salah klasifikasi, *true negative* adalah piksel yang berada di bawah *threshold* dan klasifikasinya benar, *false positive* adalah piksel di atas *threshold* dan klasifikasinya benar, dan *false negative* adalah piksel yang berada di bawah *threshold* dan salah klasifikasi. Hasil uji coba ditunjukkan pada **Tabel 5.10** dan **Tabel 5.11**. Dari tabel tersebut terlihat bahwa nilai *threshold* 0,3 memberikan hasil yang terbaik untuk ROI atas dan nilai *threshold* 0,1 memberikan hasil yang terbaik untuk ROI bawah.

Tabel 5.10 Hasil Uji Coba Nilai *Threshold* pada ROI Atas

<i>Threshold</i>	Citra Hasil	<i>F1 Score</i>
0,2		0,27
0,3		0,30
0,4		0,17

Tabel 5.11 Hasil Uji Coba Nilai *Threshold* pada ROI Bawah

<i>Threshold</i>	Citra Hasil	<i>F1 Score</i>
0,1		0,29
0,2		0,23
0,3		0,06

5.7 Uji Coba Level *Pruning* Pada RGDT

Uji coba ini dilakukan untuk mengetahui level *pruning* yang tepat pada RGDT sehingga dapat menghindari *overfitting*.

Decision tree yang optimal ukurannya akan menghasilkan performa klasifikasi yang terbaik.

Uji coba ini dilakukan dengan membandingkan nilai akurasi segmentasi citra menggunakan *decision tree* pada level *pruning* dari *tree* 50, 100, 200, 400. Data yang digunakan untuk ujicoba adalah sampel 10000 piksel dari data fitur seluruh ROI sebagai data *training* dan sampel 1000 piksel dari data fitur seluruh ROI sebagai data *testing*. Hasil uji coba ditunjukkan pada **Tabel 5.12**. Dari tabel tersebut terlihat bahwa level *pruning* yang memberikan akurasi segmentasi terbaik adalah 50.

Tabel 5.12 Hasil Uji Coba Level *Pruning*

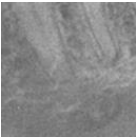
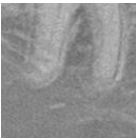
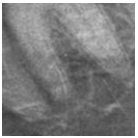
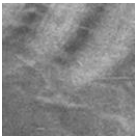
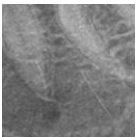
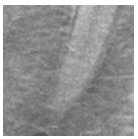
Level <i>Pruning</i>	Akurasi (%)
50	89,80
100	88,30
200	87,80
400	87,80

5.8 Uji Coba Metode Pembandingan

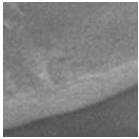
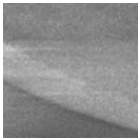
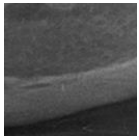
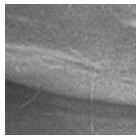
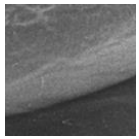
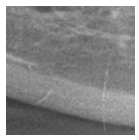
Uji coba ini dilakukan untuk mengetahui tingkat keberhasilan RGDT dalam mengklasifikasi lanjut piksel yang terdapat di atas nilai *threshold*. Uji coba dilakukan dengan membandingkan akurasi (acc), *sensitivity* (sen), dan *specificity* (spc) segmentasi citra menggunakan *K-means clustering* dan segmentasi citra menggunakan *K-means clustering* menggunakan RGDT.

Uji coba ini dilakukan dengan membandingkan hasil segmentasi masing-masing metode pada enam sampel ROI. Hasil uji coba ditunjukkan pada **Tabel 5.13** dan **Tabel 5.14**. Hasil uji coba menunjukkan bahwa RGDT dapat meningkatkan akurasi dan *sensitivity* namun menurunkan *specificity* pada ROI Atas, dan menurunkan akurasi dan *specificity* namun meningkatkan *sensitivity* pada ROI Bawah. Hasil uji coba ditampilkan pada Lampiran.

Tabel 5.13 Hasil Uji Coba Metode Pembanding pada ROI Atas

ROI	Citra ROI	Segmentasi <i>K-means Clustering</i>			Segmentasi <i>K-means Clustering</i> dan RGDT		
		acc (%)	sen (%)	spc (%)	acc (%)	sen (%)	spc (%)
ROI Bawah 1		76,81	73,57	93,73	81,80	80,48	88,72
ROI Bawah 2		91,27	91,47	90,78	91,42	95,66	80,71
ROI Bawah 3		79,64	94,00	64,44	74,61	98,77	50,70
ROI Bawah 4		89,53	91,86	85,22	89,92	95,13	80,27
ROI Bawah 5		85,33	82,58	91,34	89,23	93,38	79,68
ROI Bawah 6		88,21	87,32	91,82	92,70	93,55	89,25
Rata-Rata		85,13	86,80	86,22	86,61	92,83	78,22

Tabel 5.14 Hasil Uji Coba Metode Pembandingan pada ROI Bawah

ROI	Citra ROI	Segmentasi <i>K-means Clustering</i>			Segmentasi <i>K-means Clustering</i> dan RGDT		
		acc (%)	sen (%)	spc (%)	acc (%)	sen (%)	spc (%)
ROI Atas 1		84,37	74,12	99,67	85,63	76,32	99,51
ROI Atas 2		84,28	70,68	99,69	83,10	69,06	99,01
ROI Atas 3		95,35	91,00	100,00	89,95	97,47	91,37
ROI Atas 4		92,00	81,88	98,96	90,70	79,86	98,15
ROI Atas 5		92,25	81,80	99,23	94,67	86,96	99,82
ROI Atas 6		93,45	88,48	99,97	91,87	89,95	94,39
Rata-Rata		90,28	81,33	99,59	89,32	83,27	97,04

5.9 Evaluasi

Pada subbab ini akan dijelaskan hasil dari serangkaian uji coba yang dilakukan dan kendala yang dihadapi selama proses pengerjaan. Evaluasi yang dilakukan adalah pada tahap ekstraksi fitur, yaitu pada filter Gabor dan informasi spasial. Evaluasi pada tahap klasifikasi ulang adalah nilai *threshold* jarak piksel dengan *centroid* dan level *pruning* dari *decision tree*.

5.9.1 Evaluasi Uji Coba Parameter Filter Gabor

Pada proses ekstraksi fitur menggunakan filter Gabor, terdapat lima buah parameter yang diuji coba untuk mendapatkan nilai yang optimal, yaitu *wavelength*, *bandwith*, *gamma* dan orientasi sudut. Parameter *phase offset* telah diuji coba namun tidak memberikan efek yang signifikan terhadap akurasi *cluster* sehingga pada tugas akhir ini nilainya ditetapkan sebagai 0^0 .

Masing-masing evaluasi hasil uji coba dijelaskan lebih lanjut sebagai berikut:

5.9.1.1 Evaluasi Uji Coba Parameter Orientasi

Pada pengujian parameter orientasi, ditemukan orientasi objek pada citra ROI adalah 60^0 dan 90^0 untuk ROI atas serta 0^0 , 30^0 , dan 150^0 pada ROI bagian bawah. Pada ROI dilakukan filter Gabor dengan seluruh parameter orientasi yang didapat. Hal ini dilakukan karena orientasi objek pada ROI yang akan diproses tidak dapat ditentukan terlebih dahulu. Parameter yang didapat membuat struktur detail *trabecular bone* yang arahnya berlawanan dengan struktur akar gigi dan *cortical bone* menjadi bernilai kecil karena melewati bagian lembah pada filter Gabor.

Kelemahan yang didapat adalah, ketika terdapat struktur detail *trabecular bone* yang arahnya sejajar dengan orientasi filter Gabor maka piksel pada daerah tersebut nilainya menjadi tinggi. Hal ini umumnya terjadi pada ROI bagian bawah karena orientasi objek *cortical bone* pada bagian ini mirip dengan orientasi struktur detail *trabecular bone*.

5.9.1.2 Evaluasi Uji Coba Parameter *Wavelength*

Wavelength yang menentukan panjang gelombang kosinus pada filter Gabor turut mempengaruhi bentuk dari kernel filter Gabor. Semakin panjang gelombang yang dibentuk maka semakin banyak pula piksel yang melewati daerah nonlembah dari kernel Gabor. Hal ini mengakibatkan semakin kecil *wavelength* filter Gabor akan semakin tidak peka terhadap detail struktur, sedangkan semakin tinggi *wavelength* akan semakin peka terhadap detail struktur. Nilai *wavelength* yang terlalu tinggi akan mengakibatkan piksel-piksel *noise* menjadi tampak terang pada citra keluaran, yang dalam tugas akhir ini berarti memperterang detail struktur *trabecular bone* yang telah dikaburkan pada tahap *preprocessing*. Selain itu pada *wavelength* rendah, objek yang searah dengan orientasi namun lebarnya relatif tipis akan mendapat nilai yang rendah, sebaliknya bila lebar objek tersebut relatif lebar, maka akan mendapat nilai yang tinggi. Hal ini mengakibatkan hasil *clustering* tetap baik meskipun pada ROI dengan sudut 60^0 dilakukan pula filter Gabor dengan orientasi sudut 120^0 .

Pada pengujian parameter *wavelength*, didapatkan bahwa dua nilai yang terbaik adalah 1 dan 3. Karena filter Gabor dengan nilai *wavelength* berbeda akan menghasilkan data fitur yang berbeda pula, maka kurang representatif apabila melakukan segmentasi dengan data fitur dari keluaran satu filter Gabor saja. Oleh karena itu, ditentukan dibentuk tiga buah filter Gabor dengan nilai parameter *wavelength* yang digunakan adalah 1, 2, dan 3. Filter dibentuk sebanyak tiga buah untuk mengantisipasi piksel yang nilainya terlalu rendah *wavelength* rendah maupun piksel yang nilainya terlalu tinggi pada *wavelength* tinggi dengan tiap piksel memiliki nilai pada tiap level *wavelength* berbeda.

5.9.1.3 Evaluasi Uji Coba Parameter *Bandwith*

Pada pengujian parameter *bandwith*, didapatkan bahwa nilai yang paling optimal adalah 2. Parameter *bandwith* menentukan banyaknya fase kosinus yang tercakup dalam filter Gabor, sehingga semakin rendah nilainya, semakin peka terhadap detail struktur *trabecular bone*. Uji coba dengan nilai 0 (tidak ada fase

kosinus yang tercakup pada filter Gabor) memperlihatkan hasil clustering yang peka terhadap *noise* pada ROI atas dan hasil clustering yang tidak representatif pada ROI bawah karena kontras antarobjek yang rendah.

Uji coba dengan nilai *bandwidth* lebih besar dari 0 menunjukkan hasil *clustering* condong ke arah orientasi filter Gabor dan arah *motion blur* pada *preprocessing*. Semakin tinggi nilai *bandwidth* maka semakin tinggi daerah bukit dan semakin rendah daerah lembah pada filter Gabor sehingga objek yang searah dengan orientasi namun lebarnya relatif tipis akan mendapat nilai yang rendah, sebaliknya bila lebar objek tersebut relatif lebar, maka akan mendapat nilai yang tinggi. Hal ini mengakibatkan hasil *clustering* tetap baik meskipun pada ROI dengan sudut 60^0 dilakukan pula filter Gabor dengan orientasi sudut 120^0 . Selain itu, nilai *wavelength* yang tinggi lebih dapat mengabaikan detail struktur *trabecular bone*.

5.9.1.4 Evaluasi Uji Coba Parameter Gamma

Pada pengujian parameter *gamma*, didapatkan nilai yang paling optimal adalah 2 untuk ROI atas dan 3 untuk ROI bawah. *Gamma* mempengaruhi elipsitas dari filter Gabor. Pada citra keluaran, hal ini memberikan akibat objek-objek pada citra seperti akar gigi dan *cortical bone* menjadi lebih tinggi. Jika nilai *gamma* terlalu tinggi maka dapat mempertinggi nilai piksel detail struktur *trabecular bone*.

5.9.2 Evaluasi Uji Coba Fitur Spasial

Pada pengujian fitur spasial, didapatkan bahwa ukuran *window* terbaik untuk mendapatkan informasi spasial adalah 7×7 . Ukuran *window* yang besar mengakibatkan lebih banyak informasi nilai-nilai piksel tetangga yang tersimpan pada data fitur. Hal ini mengakibatkan berkurangnya *noise* pada hasil *clustering* dan lebih halusny bagian tepi pada gambar biner hasil *cluster*. Selain itu, informasi spasial juga memberi informasi mengenai keterkaitan nilai piksel tersebut dengan tetangganya pada saat *clustering*, sehingga suatu piksel tidak dikluster berdasar nilai piksel itu sendiri saja tetapi juga mempertimbangkan piksel-piksel

tetangganya. Pada penerapannya informasi spasial memberikan efek piksel yang terletak di tepi citra menjadi condong terklasifikasi pada satu *cluster*, yang terjadi walaupun dilakukan *zero padding* maupun *replication* pada tepi matriks. Namun hal ini tidak memberikan banyak masalah karena piksel pada tepi citra ini akan teridentifikasi sebagai piksel yang harus diklasifikasi ulang pada proses selanjutnya.

Penambahan informasi koordinat sumbu Y pada ROI bawah juga meningkatkan hasil akurasi dengan baik. Tanpa informasi sumbu Y, sering terjadi salah klasifikasi pada bagian *cortical bone* karena kontras citra ROI bawah yang rendah. Hal ini dapat diminimalisir dengan informasi sumbu Y karena tiap kluster pada sumbu Y (*trabecular bone*, *cortical bone*, dan area di bawah *cortical bone*) mempunyai nilai sumbu Y yang mirip untuk tiap kluster. Kelemahannya adalah apabila struktur *cortical bone* mempunyai orientasi yang besar, maka informasi sumbu Y cenderung membuat kesalahan pada clustering antara kluster *cortical bone* dan kluster lainnya. Tetapi hal ini dapat diatasi karena piksel yang salah klasifikasi pada sumbu Y akan masuk kedalam piksel yang harus diklasifikasi ulang pada proses selanjutnya

5.9.3 Evaluasi Uji Coba Nilai *Threshold*

Pada uji coba nilai *threshold* untuk menentukan suatu piksel terlalu jauh dari *centroid* klusternya sehingga dapat dikatakan keanggotaannya pada *cluster* tersebut masih ambigu terhadap *cluster* lain. Didapatkan nilai yang paling optimal adalah 0,3 untuk ROI atas dan 0,1 untuk ROI bawah. Diambil nilai *threshold* yang memberikan nilai *F1 Score* yang tertinggi sehingga nilai *threshold* tersebut dapat disimpulkan memiliki kemampuan terbaik dalam mendapatkan piksel yang salah klasifikasi.

Kelemahan pada penentuan nilai *threshold* adalah apabila sebuah piksel jaraknya jauh dari *centroid* kluster dimana dia menjadi anggota, tetapi pada jarak piksel itu terhadap *centroid* kluster lain juga jauh, berarti piksel tersebut tidak salah klasifikasi, karena keanggotaannya tidak ambigu terhadap kluster lain.

Namun, karena yang digunakan sebagai batas adalah nilai *threshold* semata, maka piksel tersebut tetap masuk di atas nilai *threshold*.

5.9.4 Evaluasi Uji Coba Level *Pruning* pada *Decision Tree*

Pada uji coba level *pruning* terhadap *decision tree* didapatkan bahwa nilai yang paling efektif adalah 50. Level *pruning* yang terlalu tinggi mengakibatkan *decision tree* mengalami *overfitting*. Hal ini terlihat pada hasil uji coba bahwa semakin tinggi level *pruning* maka semakin turun nilai akurasi.

5.9.5 Evaluasi Uji Coba Metode Pembandingan

Pada uji coba dengan metode pembandingan, didapatkan bahwa pada ROI atas RGDT dapat meningkatkan akurasi dan *sensitivity*, sedangkan pada ROI bawah ROI menurunkan akurasi namun dapat meningkatkan *sensitivity*. RGDT dianggap berhasil menghindarkan kesalahan pada hasil *clustering* karena mampu meningkatkan *sensitivity* yang merupakan indikator keberhasilan sistem dalam mendeteksi area *trabecular bone* pada citra.

Pada ROI Bawah terjadi penurunan akurasi karena pada RGDT, atribut yang digunakan adalah nilai fitur Gabor saja tanpa atribut koordinat sumbu Y. Hal ini berakibat, struktur *trabecular bone* yang terlihat terang pada hasil filter gabor akan masuk kedalam kelas non-*trabecular bone*. Pada tahap *clustering*, struktur *trabecular bone* tersebut berhasil terklasifikasi menjadi area *trabecular bone* karena koordinat sumbu Y piksel pada struktur *trabecular bone* tersebut dan koordinat sumbu Y titik pusat *cluster* area *trabecular bone* berdekatan. Namun, hal ini tidak menjadi masalah karena struktur area *trabecular bone* yang salah klasifikasi tersebut merupakan *region* yang luasnya kecil, sehingga dapat dibetulkan pada tahap *postprocessing*. Selain itu, pada tahap ini RGDT berhasil memperbaiki batas antara area *cortical bone* dan area *trabecular bone* sehingga *sensitivity* hasil segmentasi meningkat.

5.9.6 Evaluasi Hasil Segmentasi Metode

Evaluasi dilakukan pada 30 buah citra ROI uji coba. Pada tahap ini dipakai parameter yang telah didapatkan pada uji coba sebelumnya. Parameter tersebut adalah parameter pada filter Gabor (sudut ROI atas 60, 120 dan ROI bawah 0,150,30, parameter *wavelength* 1, 2, 3, , *gamma* 2 untuk ROI atas dan 3 untuk ROI bawah, *bandwith* adalah 2), ukuran *window* data fitur spasial sebesar 7x7, nilai *threshold* 0,3 untuk ROI Atas dan 0,1 untuk ROI bawah dan level *pruning* 50. Hasil segmentasi pada data sampel ditunjukkan pada Tabel 5.15. Hasil segmentasi masing-masing ROI ditampilkan pada bagian lampiran.

Tabel 5.15 Hasil segmentasi pada ROI

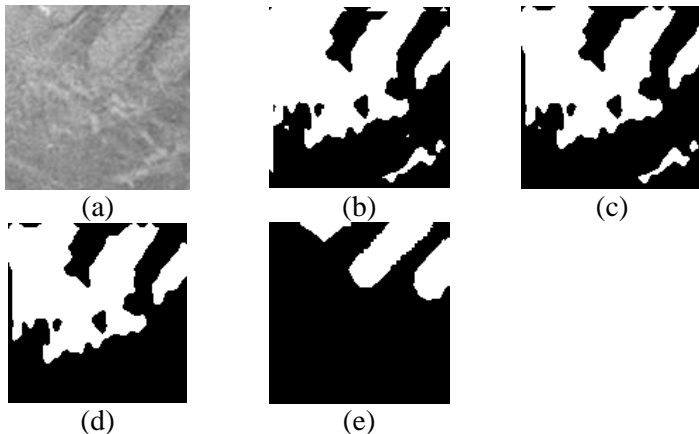
ROI	Rata-rata Akurasi (%)	Rata-rata Sensitivity (%)	Rata-rata Specificity (%)
ROI Atas	91,23	95,57	82,64
ROI Bawah	93,81	87,77	99,17

Dari evaluasi kinerja segmentasi pada ROI didapatkan bahwa rata-rata akurasi, *sensitivity*, dan *specificity* pada ROI atas adalah 91,23%, 95,57%, dan 82,64% sedangkan pada ROI bawah adalah 93,81%, 87,77%, dan 99,17%. Performa segmentasi keseluruhan ROI didapat dari rata-rata nilai akurasi, *sensitivity*, dan *specificity* ROI Atas dan ROI Bawah yaitu 92,52%, 91,67%, dan 90,90%. Nilai akurasi menunjukkan kemampuan metode dalam melakukan segmentasi, nilai *sensitivity* menunjukkan kemampuan metode dalam mensegmentasi area *trabecular bone* dengan benar, dan nilai *specificity* menunjukkan kemampuan metode dalam mensegmentasi area non-*trabecular bone* dengan benar.

Hasil segmentasi terburuk ROI atas adalah pada ROI 13 dengan nilai akurasi, *sensitivity*, dan *specificity* masing-masing

65,74%, 63,19%, dan 81,11% yang ditunjukkan pada **Gambar 5.12**. Hal ini diakibatkan karena rendahnya kontras antara objek akar gigi dengan area *trabecular bone*. Hal ini disebabkan oleh tidak meratanya pencahayaan pada citra sehingga pada ROI 14, di sekitar ujung akar gigi banyak terdapat piksel pada area *trabecular bone* yang memiliki intensitas piksel hampir sama dengan nilai intensitas piksel pada area akar gigi, yang juga sulit dibedakan secara manual oleh pengamat. Hal ini berakibat pada hasil segmentasi, banyak area *trabecular bone* yang terklasifikasi menjadi area akar gigi.

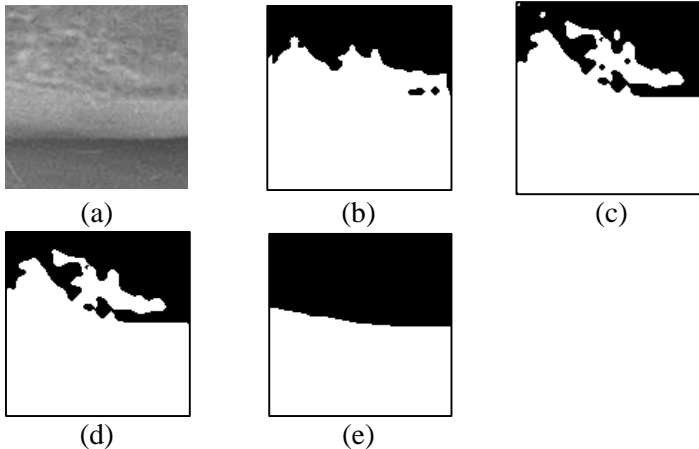
Karena daerah yang pencahayaannya tidak merata ini berupa area yang cukup luas dan tidak berupa garis, maka permasalahan ini tidak dapat diatasi pada tahap *preprocessing*, ekstraksi fitur dan *postprocessing*.



Gambar 5.12 ROI Atas terburuk. (a) ROI, (b) hasil clustering, (c) hasil klasifikasi, (d) hasil *postprocessing*, (e) *ground truth*

Sedangkan pada ROI bawah, hasil segmentasi terburuk adalah pada ROI 25 dengan nilai akurasi, *sensitivity*, dan *specificity* masing-masing 84,20%, 66,83%, dan 100,00% yang ditunjukkan pada **Gambar 5.13**. Hal ini diakibatkan karena adanya struktur *trabecular bone* pada area *trabecular bone* yang tebal dan

orientasinya mirip dengan orientasi *cortical bone*. Hal ini mengakibatkan permasalahan tersebut tidak mampu ditangani oleh tahap *preprocessing* maupun ekstraksi fitur.



Gambar 5.13 ROI Bawah terburuk. (a) ROI, (b) hasil clustering, (c) hasil klasifikasi, (d) hasil *postprocessing*, (e) *ground truth*

Selain itu, lokasi struktur *trabecular bone* yang tebal tersebut berada sangat dekat dengan area *cortical bone*, hal ini mengakibatkan perhitungan fitur spasial menjadi tidak representatif pada keadaan objek pada citra sebenarnya. Dengan nilai intensitas piksel yang mirip dengan area *cortical bone* dan lokasinya yang berdekatan dengan area *cortical bone*, nilai fitur spasial piksel pada struktur *trabecular bone* yang tebal tersebut bernilai mirip dengan piksel pada area *cortical bone* yang berada di dekatnya. Hal ini mendorong *K-means clustering* untuk mengklasifikasikannya sebagai kelas *cortical bone*.

Klasifikasi yang berbasis nilai filter Gabor tidak dapat memperbaiki kesalahan tersebut karena nilai fitur Gabor piksel pada struktur *trabecular bone* yang tebal tersebut mirip dengan area *cortical bone* karena orientasinya yang sama. *Postprocessing* tidak dapat memperbaiki kesalahan tersebut karena jarak struktur

trabecular bone yang begitu dekat sehingga pada hasil citra klasifikasi, piksel-piksel pada area struktur *trabecular bone* yang tebal tersebut tersegmentasi menyatu dengan area *cortical bone*.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah. Selain itu juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Metode *K-means clustering* dan *Root-Guided Decision Tree (RGDT)* berhasil digunakan untuk segmentasi area *trabecular bone* dengan rata-rata akurasi, *sensitivity* dan *specificity* masing-masing sebesar 92,52%, 91,67%, dan 90,90%.
2. Ekstraksi fitur yang optimal dari *dental panoramic radiograph* dapat dilakukan dengan filter Gabor (dengan parameter sudut ROI atas 60⁰, 120⁰ dan ROI bawah 0⁰, 150⁰, 30⁰, parameter *wavelength* 1, 2, 3, *gamma* 2 untuk ROI atas dan 3 untuk ROI bawah, *bandwidth* adalah 2), data fitur spasial dengan ukuran *window* 7x7, dan data koordinat sumbu Y khusus untuk ROI bawah
3. RGDT memberikan hasil yang optimal pada *threshold* jarak piksel ke *centroid* sebesar 0,3 untuk ROI atas dan 0,1 untuk ROI bawah, dan level *pruning decision tree* 50.
4. RGDT dapat memperbaiki kesalahan klasifikasi piksel pada *K-means clustering* dengan meningkatkan akurasi dan *sensitivity* sebesar 1,48% dan 6,03% pada ROI atas, dan meningkatkan *sensitivity* sebesar 1,94% pada ROI bawah.




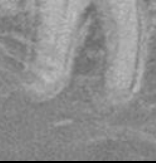


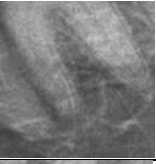


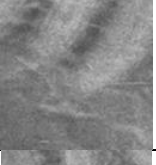





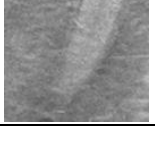


6.2 Saran

Saran yang dapat diberikan dalam pengujian metode segmentasi area *trabecular bone* adalah sebagai berikut:

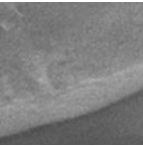


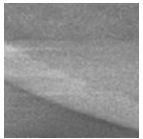


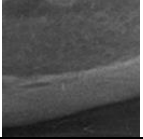


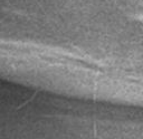


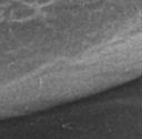


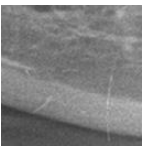


1. Perbaiki *preprocessing* untuk menyamakan detail struktur *trabecular bone* yang terlalu dominan ukuran dan intensitasnya, berorientasi mirip dengan objek pada citra, dan/atau terlalu dekat lokasinya dengan objek *non-trabecular bone* pada citra.
2. Penentuan arah orientasi filter Gabor secara otomatis dalam tahap ekstraksi fitur sehingga menghasilkan data fitur yang lebih representatif.
3. Perbaiki penentuan piksel yang dianggap ambigu pada suatu *cluster* dengan mempertimbangkan jarak piksel tersebut pada *centroid* semua *cluster*.
4. Peningkatan akurasi RGDT lewat *bootstrap aggregating* dan modifikasi algoritma untuk menurunkan *running time* program dengan meminimalisir kompleksitas penyusunan *tree*.

LAMPIRAN

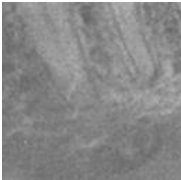


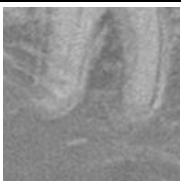





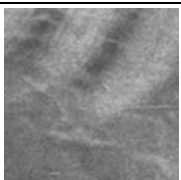


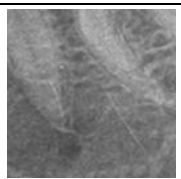


A. Hasil Uji Coba Metode Pembanding pada ROI Atas

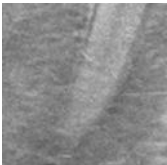


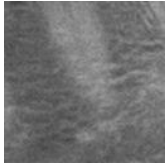


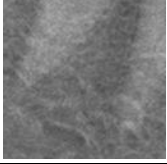


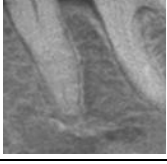


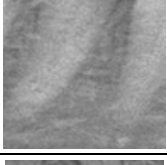


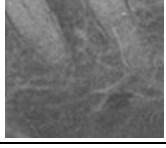


Nama ROI	Citra ROI	Hasil Segmentasi	
		Tanpa RGDT	Dengan RGDT
1			
2			
3			
4			
5			
6			

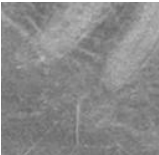


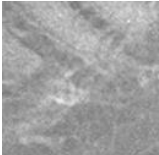


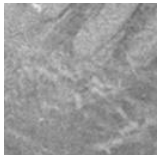


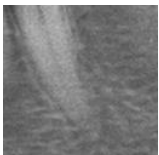


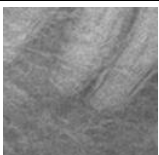


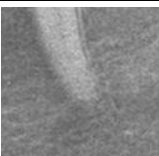


B. Hasil Uji Coba Metode Pembanding pada ROI Bawah

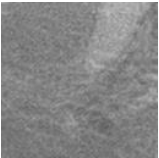


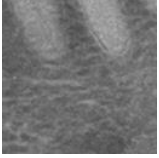


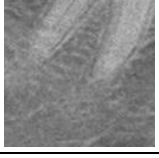


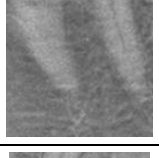


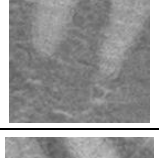





Nama ROI	Citra ROI	Hasil Segmentasi	
		Tanpa RGDT	Dengan RGDT
1			
2			
3			
4			
5			
6			

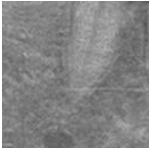


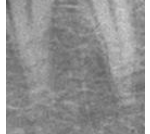


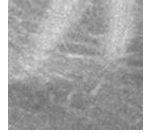


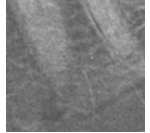


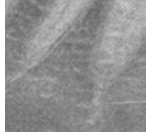


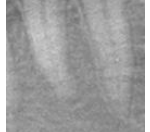


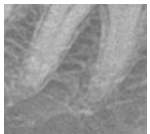


C. Hasil Segmentasi ROI Atas

Nama ROI	Citra ROI	Hasil Segmentasi	<i>Ground Truth</i>
1			
2			
3			
4			
5			

Nama ROI	Citra ROI	Hasil Segmentasi	<i>Ground Truth</i>
6			
7			
8			
9			
10			
11			

Nama ROI	Citra ROI	Hasil Segmentasi	<i>Ground Truth</i>
12			
13			
14			
15			
16			
17			

Nama ROI	Citra ROI	Hasil Segmentasi	<i>Ground Truth</i>
18			
19			
20			
21			
22			
23			

Nama ROI	Citra ROI	Hasil Segmentasi	<i>Ground Truth</i>
24			
25			
26			
27			
28			
29			
30			

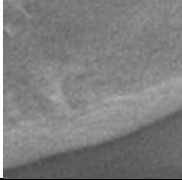


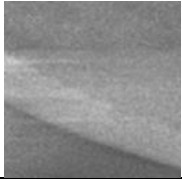


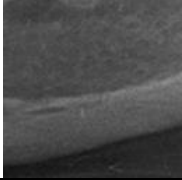


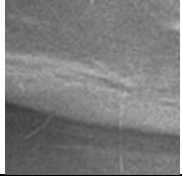


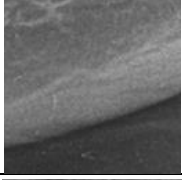


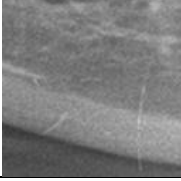


D. Performa Segmentasi ROI Atas

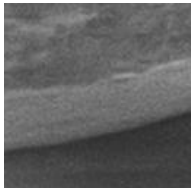


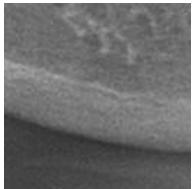


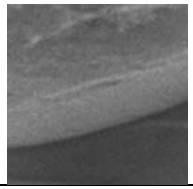


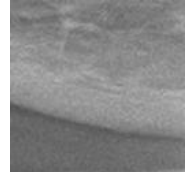


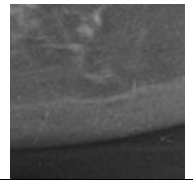


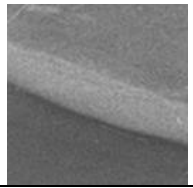


Nama ROI	Clustering			Klasifikasi			Postprocessing		
	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity
1	76,81	73,57	93,73	81,80	80,48	88,72	81,79	80,48	88,72
2	91,27	91,47	90,78	91,42	95,66	80,71	92,27	96,84	80,71
3	79,64	94,00	64,44	74,61	98,77	50,70	75,02	99,71	50,61
4	89,53	91,86	85,22	89,92	95,13	80,27	92,06	98,42	80,27
5	85,33	82,58	91,34	89,23	93,38	79,68	89,23	93,38	79,68
6	88,21	87,32	91,82	92,70	93,55	89,25	93,88	95,02	89,25
7	82,05	89,79	97,47	90,13	89,79	91,54	91,86	91,94	91,54
8	93,11	95,43	88,21	93,77	98,21	84,52	93,93	98,44	84,52
9	93,66	94,94	87,40	93,62	96,70	87,40	93,62	96,70	87,40
10	87,85	97,04	78,40	87,87	97,38	78,09	87,99	99,04	76,64
11	91,78	90,56	95,58	94,27	96,63	86,95	95,34	98,04	86,95
12	94,31	94,83	92,44	95,36	99,12	81,76	95,53	99,34	81,76
13	85,41	83,70	90,35	87,41	89,42	81,61	88,68	91,17	81,49
14	62,37	58,60	85,05	64,13	61,32	81,11	65,74	63,19	81,11

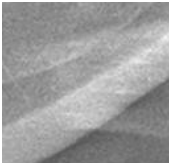


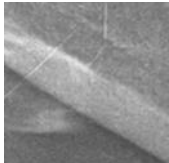





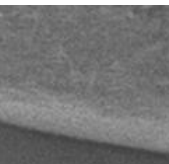


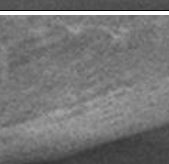


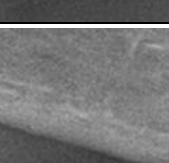


Nama	Clustering			Klasifikasi			Postprocessing		
	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity
15	95,00	97,08	86,56	96,16	99,00	84,57	96,17	99,00	84,57
16	92,35	92,94	91,26	92,23	99,19	79,49	92,27	99,27	79,46
17	96,92	97,38	93,73	98,26	99,34	90,77	98,27	99,35	90,77
18	94,54	94,04	98,59	98,67	98,94	96,46	99,20	99,53	96,46
19	95,46	97,17	87,45	95,32	98,40	80,85	94,99	98,40	78,97
20	93,40	92,76	95,93	96,54	97,53	92,65	96,62	97,62	92,65
21	93,00	96,10	87,53	92,45	99,46	80,01	92,18	99,46	79,34
22	97,12	98,30	94,15	96,28	99,74	87,48	96,41	99,93	87,48
23	94,48	96,65	87,93	94,20	99,33	78,64	94,32	99,58	78,39
24	85,40	83,82	92,47	87,11	86,02	91,97	89,62	89,09	91,97
25	89,68	91,54	85,73	90,41	95,16	80,31	90,63	95,49	80,31
26	79,87	74,25	98,95	90,97	89,55	95,81	92,11	91,02	95,81
27	93,30	93,49	92,73	95,15	98,63	84,37	95,53	99,14	84,37
28	93,83	97,65	84,22	92,93	99,38	76,71	92,60	99,62	74,90
29	87,49	94,40	76,33	84,77	97,36	64,42	86,35	99,92	64,42

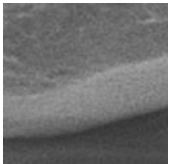


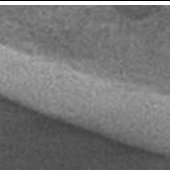


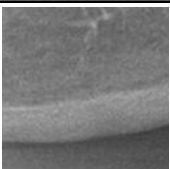


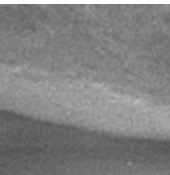


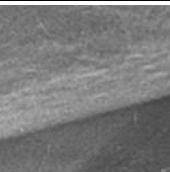


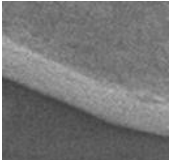


Nama ROI	Clustering			Klasifikasi			<i>Postprocessing</i>		
	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity
30	92,94	94,53	89,29	93,12	98,72	80,22	92,78	98,89	78,71
AVG	89,20	90,59	89,17	90,69	94,71	82,90	91,23	95,57	82,64
MAX	97,12	98,30	98,95	98,67	99,74	96,46	99,20	99,93	96,46
MIN	62,37	58,60	64,44	64,13	61,32	50,70	65,74	63,19	50,61

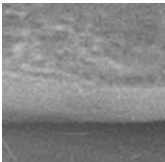


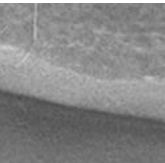


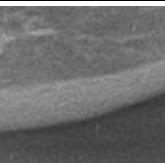


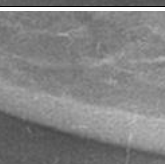


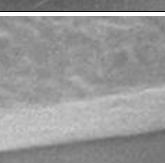


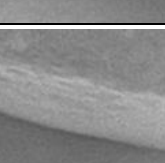


E. Hasil Segmentasi ROI Bawah

Nama ROI	Citra ROI	Hasil Segmentasi	Ground Truth
1			
2			
3			
4			
5			
6			

Nama ROI	Citra ROI	Hasil Segmentasi	Ground Truth
7			
8			
9			
10			
11			
12			

Nama ROI	Citra ROI	Hasil Segmentasi	Ground Truth
13			
14			
15			
16			
17			
18			

Nama ROI	Citra ROI	Hasil Segmentasi	Ground Truth
19			
20			
21			
22			
23			
24			

Nama ROI	Citra ROI	Hasil Segmentasi	Ground Truth
25			
26			
27			
28			
29			
30			

F. Performa Segmentasi ROI Bawah

Nama ROI	Clustering			Klasifikasi			Postprocessing		
	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity
1	84,37	74,12	99,67	85,63	76,32	99,51	87,32	78,81	100,00
2	84,28	70,68	99,69	83,10	69,06	99,01	91,98	85,04	99,84
3	95,35	91,00	100,00	89,95	97,47	91,37	89,95	100,00	79,20
4	92,00	81,88	98,96	90,70	79,86	98,15	92,88	82,72	99,87
5	92,25	81,80	99,23	94,67	86,96	99,82	95,02	87,60	99,98
6	93,45	88,48	99,97	91,87	89,95	94,39	98,27	98,42	98,08
7	97,58	94,29	99,84	91,44	79,60	99,54	97,51	93,88	100,00
8	94,43	87,21	99,87	92,27	86,47	98,40	96,79	92,55	99,99
9	85,77	72,79	96,93	82,67	65,68	97,29	86,79	71,43	100,00
10	85,67	67,30	99,85	92,32	84,06	99,54	93,38	85,91	99,91
11	85,58	72,22	100,00	87,74	77,22	99,10	89,39	79,55	100,00
12	94,62	92,31	96,07	93,80	88,64	97,03	95,89	89,30	100,00
13	91,94	84,94	98,35	83,83	70,78	95,78	95,72	91,05	100,00
14	95,92	94,00	97,17	93,77	88,05	97,47	97,98	95,94	99,31

Nama	Clustering			Klasifikasi			Postprocessing		
	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity
15	84,14	72,11	100,00	84,76	73,24	99,96	84,76	73,22	100,00
16	91,13	86,17	98,38	91,73	86,29	99,70	92,59	87,52	100,00
17	90,26	79,45	99,80	83,27	64,44	99,89	90,24	79,22	99,97
18	88,27	75,82	99,70	91,97	83,55	99,68	93,43	86,31	99,96
19	94,40	90,58	97,54	95,15	91,80	97,90	96,54	92,33	100,00
20	95,57	91,18	98,34	94,29	88,01	98,26	97,00	92,32	99,95
21	92,09	85,43	100,00	93,25	87,65	99,91	94,67	90,18	100,00
22	95,58	91,54	98,80	93,94	88,80	98,80	97,83	96,04	99,25
23	91,34	79,65	99,25	89,61	75,45	99,28	90,61	76,75	100,00
24	95,04	91,89	97,44	95,92	93,27	97,93	97,07	93,27	99,96
25	82,94	64,17	100,00	84,10	66,63	99,98	84,20	66,83	100,00
26	94,77	88,94	98,43	94,28	86,46	99,19	95,19	87,54	100,00
27	95,69	93,30	97,38	94,20	89,42	97,38	98,31	95,92	100,00
28	91,82	85,43	98,92	94,59	92,65	96,74	96,89	94,14	99,90
29	96,94	94,19	99,35	96,22	92,71	99,28	98,08	96,11	99,81

Nama ROI	Clustering			Klasifikasi			Postprocessing		
	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity	Akurasi	Sensitivity	Specificity
30	96,35	91,95	98,25	97,03	92,99	98,78	97,91	93,11	99,98
AVG	91,65	83,83	98,91	90,94	83,12	98,30	93,81	87,77	99,17
MAX	97,58	94,29	100,00	97,03	97,47	99,98	98,31	100,00	100,00
MIN	82,94	64,17	96,07	82,67	64,44	91,37	84,20	66,83	79,20

DAFTAR PUSTAKA

- [1] A. Z. Arifin, A. Asano, A. Taguchi, T. Nakamoto, M. Ohtsuka, M. Tsuda, Y. Kudo and K. Tanimoto, "Computer-aided system for measuring the mandibular cortical width on dental panoramic radiographs in identifying postmenopausal women with low bone mineral density," *Osteoporos International*, vol. 17, no. 5, pp. 753-759, May 2006.
- [2] A. Z. Arifin, M. Aryuni dan N. Yuniarti, "Informasi Tekstural untuk Identifikasi Trabecular Bone pada Citra Dental Panoramic Radiograph," *POMITS*, 2008.
- [3] A. Z. Arifin, A. Yuniarti and L. R. Dewi, "Segmentasi Trabecular Bone berdasarkan Linear Structure pada Citra Dental Panoramic Radiograph," *POMITS*, 2009.
- [4] D. A. Clausi and M. E. Jernigan, "Designing Gabor Filters for Optimal Texture Separability," *Pattern Recognition*, vol. 33, pp. 1835-1849, 2000.
- [5] S. Krinidis and V. Chartzis, "A Robust Fuzzy Local Information C-Means Clustering Algorithm," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1328-1337, 2010.
- [6] D. L. Pham, C. Xu dan J. L. Prince, "Current Methods in Image Segmentation," *Annual Review of Biomedical Engineering*, vol. 2, pp. 315-337, 2000.
- [7] M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. Rudnicka, C. G. Owen and S. A. Barman, "Retinal Vessel Segmentation using Ensemble Classifier of Bagged Decision Trees," in *IET Conference on Image Processing (IPR 2012)*, 2012.
- [8] R. GeethaRamani and L. Balasubramanian, "Retinal Blood Vessel Segmentation Employing Image Processing and Data Mining Techniques for

- Computerized Retinal Image Analysis," *Biocybernetics and Biomedical Engineering*, vol. 36, pp. 102-118, 2016.
- [9] "Radiography," US Food and Drug Administration, 6 Juni 2014. [Online]. Available: <http://www.fda.gov/Radiation-EmittingProducts/RadiationEmittingProductsandProcedures/MedicalImaging/MedicalX-Rays/ucm175028.htm>. [Diakses 13 Desember 2016].
 - [10] "Dental X-Rays," WebMD, [Online]. Available: <http://www.webmd.com/oral-health/dental-x-rays#1>. [Diakses 13 Desember 2016].
 - [11] "Panoramic Dental X-ray," Radiological Society of North America, 24 Juni 2016. [Online]. Available: <http://www.radiologyinfo.org/en/info.cfm?pg=panoramic-xray>. [Diakses 2 Januari 2017].
 - [12] G. D. Whedon and R. P. Heaney, "Bone Anatomy," Encyclopædia Britannica, 29 Januari 2013. [Online]. Available: <https://www.britannica.com/science/bone-anatomy>. [Accessed 12 Desember 2016].
 - [13] E. o. E. Britannica, "Compact Bone Anatomy," Encyclopædia Britannica, 1 Januari 2015. [Online]. Available: <https://www.britannica.com/science/compact-bone>. [Diakses 12 Desember 2016].
 - [14] E. o. E. Britannica, "Cancellous Bone Anatomy," Encyclopædia Britannica, 4 April 2015. [Online]. Available: <https://www.britannica.com/science/cancellous-bone>. [Accessed 12 Desember 2016].
 - [15] "ROI Processing in Image Processing Toolbox," MathWorks, [Online]. Available: <https://www.mathworks.com/help/images/roi-processing-in-image-processing-toolbox.html>. [Diakses 16 Desember 2016].

- [16] "Introduction to ROI Drawing," BrainMaker, [Online]. Available: http://brainimaging.waisman.wisc.edu/~oakes/spam/BrainMaker_Intro.html. [Accessed 12 Desember 2016].
- [17] F. A. Ramadhani, Segmentasi Cortical Bone Citra Dental Panoramic Radiograph menggunakan Multi-Scale Line Operator dan Supervised Classifier, Surabaya: Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, 2015.
- [18] R. C. Gonzales and R. E. Woods, Digital Image Processing Third Edition, Pearson, 2007.
- [19] R. Fisher, S. Perkins, A. Walker dan E. Wolfart, "Structuring Elements," University of Edinburgh School of Informatics, [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm>. [Diakses 18 Desember 2016].
- [20] R. Fisher, S. Perkins, A. Walker dan E. Wolfart, "Morphology - Opening," University of Edinburgh School of Informatics, [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/open.htm>. [Diakses 12 Desember 2016].
- [21] R. Fisher, S. Perkins, A. Walker dan E. Wolfart, "Digital Filters," University of Edinburgh School of Informatics, [Online]. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/filtops.htm>. [Diakses 18 Desember 2016].
- [22] R. Fisher, S. Perkins, A. Walker and E. Wolfart, "Spatial Filters - Mean Filter," University of Edinburgh School of Informatics, [Online]. Available: homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm. [Accessed 18 Desember 2016].
- [23] R. Fisher, S. Perkins, A. Walker dan E. Wolfart, "Spatial Filter - Median Filter," University of Edinburgh School of Informatics, [Online]. Available:

- <http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>.
[Diakses 12 Desember 2016].
- [24] C. V. Lab, "Gabor Filters," Department of Electrical Engineering, University of Paderborn, 17 November 1998. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TRAPP1/filter.html. [Accessed 18 Desember 2016].
- [25] D. Michie, J. S. D. and C. Taylor, Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.
- [26] M. Matteucci, "A Tutorial on Clustering Algorithms," Politecnico di Milano, [Online]. Available: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html. [Accessed 18 Desember 2016].
- [27] G. Ingargiola, "Building Classification Models: ID3 and C4.5," Temple University College of Science and Technology, [Online]. Available: <https://cis.temple.edu/~giorgio/cis587/readings/id3-c45.html>. [Diakses 18 Desember 2016].
- [28] "Confusion Matrix," Department of Computer Sciences University of Regina, [Online]. Available: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Accessed 18 Desember 2016].

BIODATA PENULIS



Ilham Gurat Adillion, lahir di Malang, pada tanggal 12 Februari 1995. Penulis menempuh pendidikan mulai dari SDN Kauman I Malang (2001-2007), MTsN Malang I (2007-2010), SMAN 3 Malang (2010-2013) hingga terakhir Institut Teknologi Sepuluh Nopember Surabaya (2013-2017) di jurusan Teknik Informatika, Fakultas Teknologi Informasi angkatan tahun 2013.

Selama belajar di kampus Teknik Informatika, penulis berkesempatan menjadi asisten dosen Komputasi Numerik pada tahun 2015. Selain mengikuti kegiatan akademik, penulis mengikuti kegiatan organisasi sekaligus mengembangkan minatnya pada bidang jurnalistik dengan berpartisipasi sebagai anggota Departemen Media Informasi HMTC (2014-2015), staff ahli Departemen Media Keluarga Muslim Informatika (2015-2016) dan ketua divisi jurnalistik Departemen Information Media BEM FTIf (2015-2016).

Penulis memiliki bidang minat Komputasi Cerdas Visi (KCV) dengan fokus studi pada bidang *image processing*, komputasi biomedik dan *data mining*. Komunikasi dengan penulis dapat melalui email: **ilhamgurata@gmail.com**.